

# 머신러닝 CHAPTER 03 객관식 퀴즈

## 회귀 알고리즘과 모델 규제

### Part 1. 이론 및 개념 (1~50번)

1. 지도 학습에서 임의의 수치를 예측하는 문제를 무엇이라 하는가?

- ① 분류(classification)
- ② 회귀(regression)
- ③ 클러스터링(clustering)
- ④ 차원 축소(dimensionality reduction)
- ⑤ 강화 학습(reinforcement learning)

2. 다음 중 회귀(regression) 문제에 해당하는 것은?

- ① 이메일이 스팸인지 판별
- ② 내년도 경제 성장률 예측
- ③ 사진 속 동물 종류 분류
- ④ 고객 이탈 여부 예측
- ⑤ 영화 장르 분류

3. k-최근접 이웃 회귀에서 새로운 샘플의 타깃값을 예측하는 방법은?

- ① 가장 가까운 이웃 하나의 타깃값을 그대로 사용
- ② 가장 가까운 k개 이웃 타깃값의 평균을 사용
- ③ 모든 훈련 데이터의 타깃값 평균을 사용
- ④ 가장 먼 이웃의 타깃값을 사용
- ⑤ 이웃 타깃값의 중앙값을 사용

4. 사이킷런에서 k-최근접 이웃 회귀 모델을 만드는 클래스는?

- ① KNeighborsClassifier
- ② KNeighborsRegressor
- ③ LinearRegression
- ④ Ridge
- ⑤ Lasso

5. 회귀 모델의 대표적인 평가 지표인 결정계수( $R^2$ )가 1에 가까울 때의 의미는?

- ① 모델 예측이 타깃 평균 수준이다
- ② 모델이 과소적합되었다

- ③ 모델의 예측이 실제 타겟에 매우 가깝다
- ④ 모델이 반드시 과대적합되었다
- ⑤ 모델의 오차가 매우 크다

6. 결정계수( $R^2$ )가 0에 가깝다는 것은 무엇을 의미하는가?

- ① 완벽한 예측 성능
- ② 모델이 타겟의 평균 정도를 예측하는 수준
- ③ 예측이 타겟과 반대 방향
- ④ 데이터에 오류가 있다
- ⑤ 모델이 학습되지 않았다

7. 결정계수( $R^2$ )의 값이 음수가 될 수 있는 경우는?

- ① 예측이 타겟보다 항상 큰 경우
- ② 예측이 타겟의 평균보다도 못한 성능일 때
- ③ 데이터가 너무 많은 경우
- ④ 특성이 하나일 때만 발생
- ⑤ 정상적인 모델에서는 절대 발생하지 않는다

8. 사이킷런의 `mean_absolute_error()` 함수가 계산하는 MAE의 의미는?

- ① 예측값과 실제값 차이의 제곱 평균
- ② 예측값과 실제값 차이의 절댓값 평균
- ③ 결정계수의 역수
- ④ 훈련 세트의 정확도
- ⑤ 특성 간의 상관계수

9. 훈련 세트 점수가 매우 높고 테스트 세트 점수가 매우 낮을 때 이 모델의 상태는?

- ① 과소적합(underfitting)
- ② 과대적합(overfitting)
- ③ 최적 상태
- ④ 데이터 누수
- ⑤ 전처리 오류

10. 테스트 세트 점수가 훈련 세트 점수보다 높거나 두 점수가 모두 낮은 경우를 무엇이라 하는가?

- ① 과대적합
- ② 과소적합
- ③ 정상 상태
- ④ 데이터 누수

⑤ 샘플링 편향

11. k-최근접 이웃에서 과소적합을 해결하려면 k값을 어떻게 조절해야 하는가?

- ① k를 늘린다
- ② k를 줄인다
- ③ k를 0으로 설정한다
- ④ k를 전체 샘플 수로 설정한다
- ⑤ k를 변경하지 않는다

12. k-최근접 이웃에서 과대적합을 해결하려면 k값을 어떻게 조절해야 하는가?

- ① k를 줄인다
- ② k를 늘린다
- ③ k를 1로 고정한다
- ④ k를 음수로 설정한다
- ⑤ k값과 무관하다

13. k-최근접 이웃에서 k값을 크게 하면 모델은 어떻게 변하는가?

- ① 더 복잡해진다
- ② 변화 없다
- ③ 더 단순해진다
- ④ 항상 과대적합된다
- ⑤ 항상 성능이 좋아진다

14. 넘파이 배열의 크기(shape)를 변경하는 메서드는?

- ① `resize()`
- ② `reshape()`
- ③ `transform()`
- ④ `convert()`
- ⑤ `reformat()`

15. `reshape(-1, 1)`에서 -1의 의미는?

- ① 행 수를 1로 설정
- ② 열 수를 자동 계산
- ③ 나머지 원소 개수에 맞춰 해당 차원을 자동 계산
- ④ 배열을 1차원으로 변환
- ⑤ 음수 값을 제거

16. 사이킷런에 사용할 훈련 세트의 입력 데이터는 어떤 형태여야 하는가?

- ① 1차원 배열
- ② 2차원 배열(행: 샘플, 열: 특성)
- ③ 3차원 배열
- ④ 딕셔너리
- ⑤ 데이터프레임만 가능

17. k-최근접 이웃 회귀가 훈련 범위를 벗어난 데이터에서 약한 근본적인 이유는?

- ① 알고리즘 자체에 버그가 있어서
- ② 가장 가까운 이웃의 타깃 평균만 사용하므로 새로운 증가 추세를 만들어 내지 못해서
- ③ 훈련 시간이 부족해서
- ④ 특성 수가 부족해서
- ⑤ 스케일링을 하지 않아서

18. 교재에서 50cm 농어와 100cm 농어를 k-최근접 이웃 회귀로 예측했을 때 비슷한 값(약 1033g)이 나온 이유는?

- ① 두 농어의 실제 무게가 같아서
- ② 훈련 범위를 벗어나면 결국 같은 이웃 그룹의 평균을 참조하므로
- ③ k값이 너무 커서
- ④ 데이터 전처리를 하지 않아서
- ⑤ 테스트 세트가 너무 작아서

19. 선형 회귀(linear regression)가 학습하는 것으로 가장 적절한 것은?

- ① 가장 가까운 이웃
- ② 특성과 타깃 사이의 관계를 가장 잘 나타내는 선형 방정식
- ③ 의사결정 트리
- ④ 확률 분포
- ⑤ 클러스터 중심

20. 선형 회귀가 k-최근접 이웃 회귀보다 외삽(extrapolation)에 강한 이유는?

- ① 더 많은 데이터를 저장하므로
- ② 관계식(방정식)을 학습했으므로 직선을 연장하여 예측할 수 있어서
- ③ 항상 더 높은 정확도를 보이므로
- ④ 내부적으로 k-최근접 이웃을 사용하므로
- ⑤ 스케일링이 필요 없으므로

21. 사이킷런에서 선형 회귀 모델을 만드는 클래스는?

- ① KNeighborsRegressor

- ② LinearRegression
- ③ PolynomialRegression
- ④ StandardScaler
- ⑤ PolynomialFeatures

22. LinearRegression 객체의 coef\_ 속성에 저장되는 것은?

- ① 절편(intercept)
- ② 학습된 특성의 계수(기울기)
- ③ R<sup>2</sup> 점수
- ④ 예측값 배열
- ⑤ 하이퍼파라미터 목록

23. LinearRegression 객체의 intercept\_ 속성에 저장되는 것은?

- ① 기울기(계수)
- ② 선형 방정식의 절편
- ③ R<sup>2</sup> 점수
- ④ 훈련 데이터의 평균
- ⑤ 하이퍼파라미터

24. 머신러닝 모델이 특성에서 학습한 파라미터(예: coef\_, intercept\_)를 무엇이라 부르는가?

- ① 하이퍼파라미터
- ② 모델 파라미터
- ③ 전처리 변수
- ④ 손실값
- ⑤ 규제 파라미터

25. 사이킷런에서 다항 회귀를 훈련할 때 사용하는 모델 클래스는?

- ① PolynomialRegression
- ② LinearRegression
- ③ PolynomialFeatures
- ④ Ridge
- ⑤ KNeighborsRegressor

26. 다항 회귀가 곡선을 학습하면서도 '선형 회귀'의 범주에 속하는 이유는?

- ① 입력 데이터가 직선이므로
- ② 계수(a, b, c 등)에 대해 선형이므로
- ③ 알고리즘이 단순하므로

- ④ 결과가 항상 직선이므로
- ⑤ 특성이 반드시 하나이므로

27. 농어의 길이를 제공하여 원래 데이터에 나란히 붙일 때 사용한 넘파이 함수는?

- ① np.append()
- ② np.column\_stack()
- ③ np.row\_stack()
- ④ np.merge()
- ⑤ np.concatenate()

28. 다중 회귀(multiple regression)란 무엇인가?

- ① 하나의 특성으로 여러 타깃을 동시에 예측하는 모델
- ② 여러 개의 특성을 사용하는 선형 회귀
- ③ 여러 모델을 앙상블하는 기법
- ④ 비선형 함수만 사용하는 회귀
- ⑤ k-최근접 이웃 회귀의 다른 이름

29. 특성 공학(feature engineering)의 정의로 올바른 것은?

- ① 새로운 데이터를 수집하는 작업
- ② 기존의 특성을 사용해 새로운 특성을 뽑아내는 작업
- ③ 모델의 하이퍼파라미터를 조정하는 작업
- ④ 테스트 세트를 만드는 작업
- ⑤ 모델을 평가하는 작업

30. 사이킷런에서 다항 특성을 자동으로 생성하는 변환기(transformer) 클래스는?

- ① StandardScaler
- ② PolynomialFeatures
- ③ LinearRegression
- ④ Ridge
- ⑤ MinMaxScaler

31. 사이킷런의 변환기(transformer) 클래스가 공통으로 제공하는 메서드 조합은?

- ① fit() + predict()
- ② fit() + transform()
- ③ fit() + score()
- ④ train() + test()
- ⑤ compile() + evaluate()

32. 사이킷런의 변환기(transformer)와 추정기(estimator)의 차이로 올바른 것은?

- ① 변환기는 데이터를 변환하고, 추정기는 학습과 예측을 수행한다
- ② 변환기는 분류만, 추정기는 회귀만 수행한다
- ③ 둘은 완전히 동일한 개념이다
- ④ 변환기는 GPU에서만, 추정기는 CPU에서만 동작한다
- ⑤ 변환기는 타겟이 반드시 필요하다

33. PolynomialFeatures에서 include\_bias=False로 설정하는 이유는?

- ① 계산 속도를 높이기 위해
- ② 사이킷런의 선형 모델이 자동으로 절편을 추가하므로 중복을 피하기 위해
- ③ 과대적합을 방지하기 위해
- ④ 다항 특성 생성을 비활성화하기 위해
- ⑤ 표준화를 위해

34. 규제(regularization)의 목적으로 올바른 것은?

- ① 훈련 속도를 높이기 위해
- ② 모델이 훈련 세트를 과도하게 학습하지 못하도록 제어하기 위해
- ③ 데이터 수를 늘리기 위해
- ④ 특성 수를 늘리기 위해
- ⑤ 테스트 세트의 크기를 줄이기 위해

35. 선형 회귀 모델에서 규제가 계수에 미치는 구체적 영향은?

- ① 계수를 더 크게 만든다
- ② 계수의 크기를 작게 만들어 보편적인 패턴을 학습하게 한다
- ③ 계수의 부호를 반전시킨다
- ④ 계수를 모두 1로 고정한다
- ⑤ 계수에 영향을 주지 않는다

36. 릿지(Ridge) 회귀의 규제 방식은?

- ① 계수의 절댓값을 기준으로 규제
- ② 계수를 제곱한 값을 기준으로 규제
- ③ 계수를 모두 0으로 강제
- ④ 특성을 랜덤으로 제거
- ⑤ 학습률을 자동 조정

37. 라쏘(Lasso) 회귀의 규제 방식은?

- ① 계수를 제곱한 값을 기준으로 규제

- ② 계수의 절댓값을 기준으로 규제
- ③ 특성의 평균을 기준으로 규제
- ④ 데이터 수를 기준으로 규제
- ⑤  $R^2$  점수를 기준으로 규제

38. 라쏘 회귀가 릿지 회귀와 구별되는 가장 큰 특징은?

- ① 항상 더 높은 성능을 보인다
- ② 일부 계수를 완전히 0으로 만들 수 있다
- ③ 규제가 없다
- ④ 분류에만 사용 가능하다
- ⑤ 하이퍼파라미터가 없다

39. 릿지와 라쏘에서 규제의 강도를 조절하는 하이퍼파라미터는?

- ① n\_neighbors
- ② degree
- ③ alpha
- ④ max\_iter
- ⑤ random\_state

40. 릿지 회귀에서 alpha 값을 크게 하면 어떤 방향으로 모델이 변하는가?

- ① 더 복잡해지고 과대적합 방향
- ② 성능이 반드시 좋아진다
- ③ 계수를 더 많이 줄여 과소적합 방향으로 유도된다
- ④ alpha와 모델 복잡도는 무관하다
- ⑤ 훈련 시간만 길어진다

41. 릿지 회귀에서 alpha 값을 너무 작게 하면 발생할 수 있는 문제는?

- ① 과소적합
- ② 과대적합
- ③ 모델이 학습되지 않음
- ④ 데이터 누수
- ⑤ 메모리 부족

42. 규제를 적용하기 전에 특성을 표준화해야 하는 이유는?

- ① 계산 속도를 높이기 위해
- ② 시각화를 위해
- ③ 특성마다 스케일이 다르면 규제가 계수를 공정하게 제어하지 못하므로

- ④ 사이킷런의 문법적 필수 요구사항이므로
- ⑤  $R^2$  점수를 인위적으로 높이기 위해

43. 사이킷런에서 특성을 표준화(평균 0, 표준편차 1)하는 변환기 클래스는?

- ① PolynomialFeatures
- ② StandardScaler
- ③ Ridge
- ④ Lasso
- ⑤ LinearRegression

44. 모델 파라미터와 하이퍼파라미터의 차이로 올바른 것은?

- ① 모델 파라미터는 사람이 지정하고, 하이퍼파라미터는 모델이 학습한다
- ② 모델 파라미터는 모델이 학습하고, 하이퍼파라미터는 사람이 사전에 지정한다
- ③ 둘은 같은 의미이다
- ④ 모델 파라미터는 변하지 않는 상수이다
- ⑤ 하이퍼파라미터는 학습 중 자동으로 조정된다

45. 다음 중 하이퍼파라미터에 해당하는 것을 모두 고른 것은?

- ① coef\_ (계수)
- ② alpha (규제 강도)
- ③ intercept\_ (절편)
- ④  $R^2$  점수
- ⑤ MAE 값

46. PolynomialFeatures의 degree 매개변수의 역할은?

- ① 학습률을 설정한다
- ② 생성할 다항 특성의 최고 차수를 지정한다
- ③ 규제 강도를 설정한다
- ④ 이웃의 수를 설정한다
- ⑤ 반복 횟수를 설정한다

47. 훈련 샘플 42개에 특성 55개를 사용했을 때 과대적합이 발생하기 쉬운 이유는?

- ① 특성 수가 짝수여서
- ② 샘플 수보다 특성 수가 많아 모델이 각 샘플을 개별적으로 외울 수 있어서
- ③ k값이 너무 작아서
- ④ 테스트 세트가 없어서
- ⑤ 표준화를 적용했기 때문에

48. 다음 코드의 출력은?

```
test_array = np.array([1,2,3,4])
test_array = test_array.reshape(2, 2)
print(test_array.shape)
```

- ① (4,)
- ② (2, 2)
- ③ (4, 1)
- ④ (1, 4)
- ⑤ 오류 발생

49. 다음 코드에서 오류가 발생하는 이유는?

```
np.array([1,2,3,4]).reshape(2, 3)
```

- ① reshape() 함수가 존재하지 않아서
- ② 원본 원소 4개를  $2 \times 3 = 6$ 개 크기로 바꿀 수 없어서
- ③ 정수 배열은 reshape할 수 없어서
- ④ 2차원 배열은 만들 수 없어서
- ⑤ 넘파이 버전 호환 문제

50. 다음 코드의 출력은?

```
train_input = np.array([10, 20, 30]).reshape(-1, 1)
print(train_input.shape)
```

- ① (3,)
- ② (1, 3)
- ③ (3, 1)
- ④ (-1, 1)
- ⑤ 오류 발생

## Part 2. 코드 및 응용 (51~100번)

51. 다음 코드에서 50cm와 100cm 농어의 예측값이 모두 약 1033g으로 비슷하게 나온다. 이 모델은 어떤 알고리즘인가?

```
print(model.predict([[50]]))  
print(model.predict([[100]]))
```

- ① LinearRegression
- ② KNeighborsRegressor
- ③ Ridge
- ④ Lasso
- ⑤ PolynomialFeatures

52. 다음 코드 실행 결과에서 39.01과 -709.02는 각각 무엇인가?

```
lr = LinearRegression()  
lr.fit(train_input, train_target)  
print(lr.coef_, lr.intercept_)  
# 출력: [39.01714496] -709.0186449535477
```

- ① R<sup>2</sup> 점수와 MAE
- ② 기울기(계수)와 절편
- ③ 훈련 점수와 테스트 점수
- ④ alpha와 degree
- ⑤ 평균과 표준편차

53. coef\_ = [39.01], intercept\_ = -709.02일 때, 길이 30cm 농어의 예측 무게(g)에 가장 가까운 값은?

- ① 약 462g
- ② 약 561g
- ③ 약 700g
- ④ 약 1033g
- ⑤ 약 -709g

54. 선형 회귀에서 길이가 매우 작은 농어(예: 5cm)에 대해 음수 무게가 예측될 수 있는 이유는?

- ① 데이터에 오류가 있어서
- ② 직선 모델( $y = ax + b$ )의 절편이 큰 음수이므로, 작은 입력에서는 결과가 음수가 될 수 있다
- ③ k-최근접 이웃의 한계 때문에
- ④ 과대적합 때문에
- ⑤ MAE가 높아서

55. 다음 코드 실행 후 train\_poly의 shape는?

```
train_poly = np.column_stack((train_input ** 2, train_input))
# train_input.shape = (42, 1)
```

- ① (42, 1)
- ② (42, 2)
- ③ (84, 1)
- ④ (42, 3)
- ⑤ (2, 42)

56. 다항 회귀에서  $lr.coef_ = [1.01, -21.6]$ ,  $lr.intercept_ = 116.05$ 일 때 학습된 방정식은?

- ① 무게 =  $1.01 \times \text{길이} - 21.6$
- ② 무게 =  $1.01 \times \text{길이}^2 - 21.6 \times \text{길이} + 116.05$
- ③ 무게 =  $-21.6 \times \text{길이}^2 + 1.01 \times \text{길이} + 116.05$
- ④ 무게 =  $116.05 \times \text{길이}^2$
- ⑤ 무게 =  $1.01 + 21.6 + 116.05$

57. 다음 코드에서 `read_csv()`가 반환하는 것은?

```
import pandas as pd
df = pd.read_csv('https://bit.ly/perch_csv_data')
```

- ① 넘파이 배열
- ② 파이썬 리스트
- ③ 판다스 데이터프레임
- ④ 사이킷런 모델 객체
- ⑤ CSV 파일의 원시 문자열

58. 3개의 특성(길이, 높이, 두께)을 `PolynomialFeatures(degree=2, include_bias=False)`로 변환하면 특성 수는?

- ① 3개
- ② 6개
- ③ 9개
- ④ 10개
- ⑤ 15개

59. 3개의 특성을 `PolynomialFeatures(degree=5, include_bias=False)`로 변환하면 특성 수는?

- ① 15개
- ② 35개
- ③ 55개
- ④ 75개
- ⑤ 100개

60. 다음 결과의 해석으로 올바른 것은?

훈련 점수: 0.9999999999991098

테스트 점수: -144.40579242684848

- ① 훌륭한 모델이므로 바로 실전에 사용 가능
- ② 과소적합이다
- ③ 심각한 과대적합이다
- ④ 데이터가 부족하지만 문제없다
- ⑤ 스케일링만 하면 해결된다

61. 다음 코드에서 StandardScaler를 훈련 세트(train\_poly)로 학습(fit)한 변환기로 테스트 세트(test\_poly)도 변환하는 이유는?

```
ss.fit(train_poly)
```

```
train_scaled = ss.transform(train_poly)
```

```
test_scaled = ss.transform(test_poly)
```

- ① 계산 편의를 위해
- ② 훈련과 테스트에 동일한 기준(평균, 표준편차)을 적용해야 공정한 평가가 가능하므로
- ③ 테스트 세트가 더 크기 때문에
- ④ 사이킷런의 문법적 필수 요구사항이므로
- ⑤ 성능을 의도적으로 낮추기 위해

62. 다음 코드의 출력은?

```
poly = PolynomialFeatures(include_bias=False)
```

```
poly.fit([[2, 3]])
```

```
print(poly.transform([[2, 3]]))
```

- ① [[2. 3.]]
- ② [[1. 2. 3. 4. 6. 9.]]
- ③ [[2. 3. 4. 6. 9.]]
- ④ [[4. 6. 9.]]
- ⑤ [[2. 3. 6.]]

63. 릿지 회귀에서 최적의 alpha 값을 찾기 위해 교재에서 사용한 방법은?

- ① 항상 alpha=1만 사용
- ② 여러 alpha 값으로 훈련/테스트 점수를 그래프로 비교하여 가장 균형 잡힌 값을 선택
- ③ alpha를 0으로 고정
- ④ alpha를 무한대로 설정
- ⑤ alpha를 랜덤으로 생성

64. alpha 값을 그래프에 표시할 때 x축을 로그 스케일로 사용하는 이유는?

- ① 데이터가 음수이므로
- ② alpha 값이 0.001에서 100까지 10배씩 변하므로, 일반 축으로는 왼쪽이 너무 촘촘해져서

- ③ 로그 스케일이 항상 정확하므로
- ④ 사이킷런의 필수 요구사항이므로
- ⑤ 그래프를 더 예쁘게 만들기 위해

65. 다음 코드의 출력값 40은 무엇을 의미하는가?

```
print(np.sum(lasso.coef_ == 0))
```

- ① 0이 아닌 계수의 합
- ② 계수가 정확히 0인 특성의 개수
- ③ 전체 계수 값의 합
- ④ MAE 값
- ⑤ alpha 값

66. 라쏘 회귀에서 55개 특성 중 40개의 계수가 0이면, 모델이 실제로 사용하는 특성 수는?

- ① 55개
- ② 40개
- ③ 15개
- ④ 0개
- ⑤ 95개

67. 라쏘 모델 훈련 시 ConvergenceWarning이 발생하는 이유는?

- ① 데이터에 결측치가 있어서
- ② 반복적인 계산이 지정한 반복 횟수 안에 충분히 수렴하지 못해서
- ③ alpha 값이 너무 크기 때문에
- ④ 특성 수가 너무 적어서
- ⑤ 사이킷런 버전이 낮아서

68. 라쏘의 ConvergenceWarning을 해결하기 위해 조정하는 매개변수는?

- ① alpha
- ② degree
- ③ max\_iter
- ④ n\_neighbors
- ⑤ random\_state

69. 다음 중 사이킷런의 모델 클래스(추정기)에서 제공하지 않는 메서드는?

- ① fit()
- ② predict()
- ③ score()
- ④ evaluate()

⑤ get\_params()

70. 다음 모델의 훈련/테스트 점수를 보고 상태를 판단하면?

훈련 세트  $R^2$ : 0.97

테스트 세트  $R^2$ : 0.98

- ① 과대적합이다
- ② 과소적합 경향이 있다
- ③ 최적 상태이다
- ④ 데이터에 오류가 있다
- ⑤ 규제가 너무 강하다

71. 사이킷런의 KNeighborsRegressor 클래스에서 n\_neighbors의 기본값은?

- ① 1
- ② 3
- ③ 5
- ④ 7
- ⑤ 10

72. 다음 코드에서 knr.kneighbors([[50]])가 반환하는 것은?

```
distances, indexes = knr.kneighbors([[50]])
```

- ① 예측값 하나
- ②  $R^2$  점수
- ③ 이웃까지의 거리 배열과 이웃 샘플의 인덱스 배열
- ④ 모델 파라미터
- ⑤ 표준화된 데이터

73. mean\_absolute\_error() 함수에 전달하는 인수의 올바른 순서는?

- ① (예측값, 실제값)
- ② (실제값, 예측값)
- ③ 순서는 무관하다
- ④ (입력 데이터, 타겟 데이터)
- ⑤ (모델 객체, 데이터)

74. a, b, c 세 특성에 PolynomialFeatures(degree=3)을 적용했을 때, 다음 중 포함되지 않는 항은?

- ① 1 (절편항)
- ②  $a \times b$
- ③  $a^2 \times b$
- ④  $a \times b^3$
- ⑤  $a^3$

75. 다음 중 릿지(Ridge) 회귀의 alpha 매개변수 기본값은?

- ① 0
- ② 0.1
- ③ 1
- ④ 10
- ⑤ 100

76. 다음 결과에서 alpha=0.1이 최적인 이유는?

Ridge(alpha=0.1) → 훈련: 0.9904, 테스트: 0.9828

- ① 규제가 전혀 없어서
- ② 훈련과 테스트 점수가 모두 높고 차이가 작아 과대·과소적합의 균형이 잡혔으므로
- ③ 훈련 점수만 높아서
- ④ 테스트 점수만 높아서
- ⑤ alpha가 가장 작은 값이어서

77. 교재의 산점도에서 x축이 농어의 길이, y축이 무게일 때 관찰되는 패턴은?

- ① 무관한 분포(랜덤)
- ② 길이가 길수록 무게가 감소하는 음의 상관관계
- ③ 길이가 길수록 무게가 증가하는 양의 상관관계
- ④ 원형 분포
- ⑤ 균일 분포

78. 머신러닝에서 '가중치(weight)'라는 용어가 선형 회귀에서 의미하는 것은?

- ① 농어의 물리적 무게
- ② 방정식의 기울기와 절편을 모두 포함하는 모델 파라미터
- ③ 하이퍼파라미터
- ④ 입력 데이터 배열
- ⑤ 손실 함수의 값

79. 다항 회귀를 사용하여 과소적합을 해결할 수 있었던 이유는?

- ① 데이터를 더 많이 수집했으므로
- ② 직선보다 유연한 곡선(2차 함수) 형태를 표현할 수 있었으므로
- ③ 규제를 적용했으므로
- ④ k값을 줄였으므로
- ⑤ 테스트 세트를 제거했으므로

80. 라쏘 회귀를 특성 선택(feature selection) 용도로 사용할 수 있는 이유는?

- ① 항상 최고 성능을 보이므로

- ② 불필요한 특성의 계수를 0으로 만들어 자동으로 특성을 제거하는 효과가 있으므로
- ③ PolynomialFeatures를 내부적으로 포함하므로
- ④ PCA를 자동으로 수행하므로
- ⑤ 교차 검증을 내장하고 있으므로

81. 교재에서 다루는 보간(interpolation)과 외삽(extrapolation)의 차이로 올바른 것은?

- ① 보간은 훈련 범위 밖 예측, 외삽은 범위 안 예측
- ② 보간은 훈련 범위 안의 예측, 외삽은 범위 밖의 예측
- ③ 둘은 같은 의미이다
- ④ 보간은 분류에만, 외삽은 회귀에만 사용
- ⑤ 보간은 규제, 외삽은 특성 공학을 의미

82. read\_csv() 함수의 sep 매개변수의 역할은?

- ① 파일 경로를 지정한다
- ② CSV 파일의 구분자를 지정한다
- ③ 읽을 행 수를 지정한다
- ④ 열 이름을 지정한다
- ⑤ 인코딩 방식을 지정한다

83. 다음 코드 실행 후 train\_poly의 shape는?

```
poly = PolynomialFeatures(degree=2, include_bias=False)
poly.fit(train_input) # train_input.shape = (42, 3)
train_poly = poly.transform(train_input)
print(train_poly.shape)
```

- ① (42, 3)
- ② (42, 6)
- ③ (42, 9)
- ④ (42, 10)
- ⑤ (42, 12)

84. get\_feature\_names\_out() 메서드의 역할은?

- ① 모델의 R<sup>2</sup> 점수를 출력한다
- ② PolynomialFeatures가 생성한 각 특성이 어떤 입력 조합인지 이름을 알려준다
- ③ 데이터를 표준화한다
- ④ 예측값 배열을 반환한다
- ⑤ 하이퍼파라미터 목록을 출력한다

85. StandardScaler 객체의 mean\_과 scale\_ 속성에 각각 저장되는 것은?

- ① 예측값과 실제값

- ②  $R^2$  점수와 MAE
- ③ 훈련 세트에서 계산한 각 특성의 평균과 표준편차
- ④ 테스트 세트의 평균과 분산
- ⑤ 모델의 계수와 절편

86. 다음 alpha별 릿지 모델 결과에서 최적의 alpha 값은?

alpha=0.001 → 훈련 0.99, 테스트 0.85

alpha=0.1 → 훈련 0.99, 테스트 0.98

alpha=100 → 훈련 0.80, 테스트 0.79

- ① 0.001 — 훈련 점수가 가장 높으므로
- ② 0.1 — 훈련·테스트 점수가 모두 높고 차이가 가장 작으므로
- ③ 100 — 두 점수의 차이가 가장 작으므로
- ④ 어떤 값이든 성능 차이 없다
- ⑤ 추가 실험 없이 판단 불가

87. 특성이 2개인 다중 회귀에서 학습되는 모델의 기하학적 형태는?

- ① 직선
- ② 곡선
- ③ 평면
- ④ 점
- ⑤ 구

88. 교재에서 라쏘 회귀가 최적의 계수를 찾기 위해 사용하는 최적화 방법은?

- ① 확률적 경사 하강법
- ② 좌표 하강법(coordinate descent)
- ③ 뉴턴 방법
- ④ 유전 알고리즘
- ⑤ 그리드 서치

89. 릿지(Ridge) 클래스의 solver 매개변수 기본값은?

- ① sag
- ② saga
- ③ auto
- ④ lbfgs
- ⑤ sgd

90. PolynomialFeatures에서 interaction\_only=True로 설정하면 어떤 항이 제외되는가?

- ① 교호항(특성 간 곱셈)
- ② 거듭제곱 항( $a^2$ ,  $b^2$  등 제곱·세제곱)

- ③ 원래 특성 자체
- ④ 절편항(1)
- ⑤ 모든 새로운 특성

91. 다음 코드에서 올바른 실행 순서를 고르시오.

가. `lr.predict([[50**2, 50]])`  
나. `lr.fit(train_poly, train_target)`  
다. `lr = LinearRegression()`

- ① 다 → 나 → 가
- ② 가 → 나 → 다
- ③ 나 → 가 → 다
- ④ 다 → 가 → 나
- ⑤ 나 → 다 → 가

92.  $R^2$ 와 MAE를 함께 확인하면 좋은 이유는?

- ① 둘은 항상 같은 값을 반환하므로 검증용이다
- ②  $R^2$ 는 비율 기반의 상대적 설명력, MAE는 실제 단위(g 등)의 오차 크기를 보여주어 다각적 평가가 가능하다
- ③ MAE가  $R^2$ 보다 항상 정확하므로 MAE만 보면 된다
- ④ 사이킷런이 두 지표를 동시에 반환하므로
- ⑤  $R^2$ 만으로는 계산이 불가능하므로

93. Ridge와 Lasso를 임포트하는 올바른 코드는?

- ① `from sklearn.preprocessing import Ridge, Lasso`
- ② `from sklearn.linear_model import Ridge, Lasso`
- ③ `from sklearn.neighbors import Ridge, Lasso`
- ④ `from sklearn.metrics import Ridge, Lasso`
- ⑤ `import Ridge, Lasso`

94. 다음 중 회귀(regression) 문제가 아닌 것은?

- ① 집값 예측
- ② 배달 도착 시간 예측
- ③ 이메일 스팸 여부 분류
- ④ 전력 사용량 예측
- ⑤ 매출 예측

95. 사이킷런 회귀 모델의 `score()` 메서드가 반환하는 지표는?

- ① 정확도(accuracy)
- ② 결정계수( $R^2$ )

- ③ MAE
- ④ MSE
- ⑤ F1 점수

96. 다음 중 과대적합과 과소적합에 대한 설명으로 틀린 것은?

- ① 과대적합인 모델은 훈련 세트의 점수가 높다
- ② 과대적합인 모델은 테스트 세트의 점수도 높다
- ③ 과소적합인 모델은 훈련 세트의 점수가 낮을 수 있다
- ④ 과소적합은 모델이 너무 단순할 때 발생할 수 있다
- ⑤ 과대적합은 모델이 훈련 데이터를 외울 때 발생한다

97. 이미 생성된 KNeighborsRegressor 객체 knr의 이웃 수를 5에서 3으로 바꾸는 올바른 코드는?

- ① knr = KNeighborsRegressor(n\_neighbors=3)
- ② knr.set\_neighbors(3)
- ③ knr.n\_neighbors = 3
- ④ knr.update(n\_neighbors=3)
- ⑤ knr.k = 3

98. 다음 코드에서 marker='D'와 marker='^'는 각각 무엇을 표시하기 위해 사용되었는가?

```
plt.scatter(train_input[indexes], train_target[indexes], marker='D')
plt.scatter(50, 1033, marker='^')
```

- ① D는 전체 훈련 데이터, ^는 테스트 데이터
- ② D는 50cm 농어, ^는 이웃 샘플
- ③ D는 이웃 샘플, ^는 예측 대상인 50cm 농어
- ④ D는 테스트 데이터, ^는 훈련 데이터
- ⑤ D는 절편, ^는 기울기를 나타낸다

99. 다음 코드에서 Lasso(alpha=alpha, max\_iter=10000)으로 max\_iter를 10000으로 설정한 이유는?

```
for alpha in [0.001, 0.01, 0.1, 1, 10, 100]:
    lasso = Lasso(alpha=alpha, max_iter=10000)
    lasso.fit(train_scaled, train_target)
```

- ① 훈련 데이터를 10000번 반복 사용하기 위해
- ② 라쏘의 좌표 하강법이 충분히 수렴하도록 반복 횟수를 늘리기 위해
- ③ alpha 값의 범위를 10000까지 확장하기 위해
- ④ 특성 수를 10000개로 늘리기 위해
- ⑤ 테스트 세트를 10000번 평가하기 위해

100. 다항 회귀 모델에 50cm 농어를 예측시킬 때 입력을 [[50\*\*2, 50]] 형태로 넣어야 하는 이유는?

```
# 모델은 np.column_stack((train_input**2, train_input))으로 만든 train_poly로 훈련됨
```

- ① 50을 두 번 넣어야 정확도가 높아지므로

- ② 훈련 시 (길이<sup>2</sup>, 길이) 순서로 특성을 구성했으므로, 예측 시에도 동일한 구조로 입력해야 하므로
- ③ 사이킷런의 predict()가 항상 2개 이상의 값을 요구하므로
- ④ 50의 제곱근과 50을 함께 넣어야 다항 회귀가 되므로
- ⑤ 2차원 배열을 만들기 위해 아무 값이나 하나 더 추가한 것이다

# 정답 및 해설

## 1번 정답: ②

[해설] 분류는 '도미나 빙어냐'처럼 클래스를 예측하는 문제이고, 회귀는 '무게가 몇 g인가'처럼 임의의 수치를 예측하는 문제이다. 교재에서 '회귀는 정해진 클래스가 없고 임의의 수치를 출력'한다고 명시하고 있다.

## 2번 정답: ②

[해설] 경제 성장률은 연속적인 수치이므로 회귀 문제이다. 교재에서도 '내년도 경제 성장률을 예측하거나 배달이 도착할 시간을 예측'하는 것을 회귀의 대표 예시로 들고 있다. 나머지는 모두 클래스를 예측하는 분류 문제이다.

## 3번 정답: ②

[해설] k-최근접 이웃 회귀는 예측하려는 샘플에 가장 가까운 k개의 이웃을 찾고, 그 이웃 샘플들의 타깃값을 평균하여 예측값으로 사용한다. 교재 예시에서 이웃의 타깃값이 100, 80, 60이면 평균인 80이 예측값이 된다.

## 4번 정답: ②

[해설] KNeighborsRegressor가 k-최근접 이웃 회귀를 구현한 클래스이다. KNeighborsClassifier는 분류용 클래스이고, LinearRegression·Ridge·Lasso는 선형 회귀 계열 클래스이다.

## 5번 정답: ③

[해설]  $R^2$ 의 공식에서 예측이 타깃에 가까워지면 분자(타깃-예측)<sup>2</sup>의 합이 0에 가까워져  $R^2$ 가 1에 접근한다. 따라서 1에 가까울수록 예측 성능이 좋다는 뜻이다. 단, 1에 가깝다고 해서 반드시 좋은 모델인 것은 아니며, 과대적합 여부는 훈련/테스트 점수를 함께 봐야 판단할 수 있다.

## 6번 정답: ②

[해설]  $R^2$  공식에서 분자(타깃-예측)<sup>2</sup>의 합과 분모(타깃-평균)<sup>2</sup>의 합이 비슷하면  $R^2$ 는 0에 가까워진다. 이는 모델이 단순히 타깃의 평균 정도를 예측하는 것과 다름없다는 의미이다.

## 7번 정답: ②

[해설]  $R^2$  공식에서 분자(타깃-예측)<sup>2</sup>의 합이 분모(타깃-평균)<sup>2</sup>의 합보다 크면  $R^2$ 가 음수가 된다. 이는 모델의 예측이 단순히 평균만 예측하는 것보다도 나쁘다는 의미이다. 교재에서도 그래프 ②에서 예측을 반대로 하면  $R^2$ 가 음수가 될 수 있다고 설명하고 있다.

## 8번 정답: ②

[해설] MAE(Mean Absolute Error)는 예측값과 실제값 차이의 절댓값을 평균한 것이다. 교재 예제에서 MAE가 약 19g으로 나왔는데, 이는 예측이 평균적으로 19g 정도 실제 무게와 차이가 난다는 뜻이다.  $R^2$ 가 비율 기반 지표라면 MAE는 실제 단위(g)로 해석할 수 있는 절대적 오차 지표이다.

## 9번 정답: ②

[해설] 과대적합은 모델이 훈련 세트에 과도하게 맞춰져서, 훈련 세트에서는 점수가 굉장히 좋지만 테스트

트 세트에서는 점수가 굉장히 나쁜 경우이다. 교재에서 55개 특성으로 42개 샘플을 훈련했을 때 훈련 점수 0.9999, 테스트 점수 -144.40이 나온 것이 대표적인 과대적합 사례이다.

#### 10번 정답: ②

[해설] 교재에서 과소적합을 '훈련 세트보다 테스트 세트의 점수가 높거나 두 점수가 모두 너무 낮은 경우'로 정의하고 있다. 모델이 너무 단순하여 훈련 세트에 적절히 훈련되지 않은 상태를 말한다.

#### 11번 정답: ②

[해설] 교재에서 '과소적합일 경우 모델을 더 복잡하게 만들어야 하며, k-최근접 이웃의 경우 k 값을 줄인다'고 명시하고 있다. k를 줄이면 국지적인 패턴에 민감해져 모델이 더 복잡해지고 훈련 데이터에 더 잘 맞게 된다.

#### 12번 정답: ②

[해설] 교재에서 '과대적합일 경우 모델을 덜 복잡하게 만들어야 하며, k-최근접 이웃의 경우 k 값을 늘린다'고 명시하고 있다. k를 늘리면 더 넓은 범위의 평균을 참고하므로 데이터 전반의 일반적인 패턴을 따르게 되어 모델이 단순해진다.

#### 13번 정답: ③

[해설] 이웃의 개수를 늘리면 데이터 전반에 있는 일반적인 패턴을 따르게 되므로 모델이 단순해진다. 반대로 이웃의 개수를 줄이면 훈련 세트에 있는 국지적인 패턴에 민감해져 모델이 복잡해진다.

#### 14번 정답: ②

[해설] reshape() 메서드는 배열의 크기를 바꿔주는 넘파이 기능이다. 크기가 바뀐 새로운 배열을 반환하며, 지정한 크기의 원소 총 수가 원본 배열의 원소 수와 일치해야 한다. 일치하지 않으면 에러가 발생한다.

#### 15번 정답: ③

[해설] 교재에서 '크기에 -1을 지정하면 나머지 원소 개수로 모두 채우라는 의미'라고 설명하고 있다. 예를 들어 train\_input.reshape(-1, 1)은 '열 수는 1로 고정하고, 행 수는 전체 원소 수에 맞춰 자동 계산하라'는 뜻이다. 샘플 수를 굳이 사람이 세지 않아도 되어 코드가 유연해진다.

#### 16번 정답: ②

[해설] 교재에서 '사이킷런에 사용할 훈련 세트는 2차원 배열이어야 함'이라고 명시하고 있다. 특성이 하나뿐이더라도 (샘플 수, 1) 형태의 2차원 배열을 만들어야 한다. 사이킷런이 일관된 인터페이스를 유지하기 위해서이다.

#### 17번 정답: ②

[해설] k-최근접 이웃 회귀는 새로운 법칙을 세우지 못하고, 가장 가까운 기존 샘플 몇 개를 끌어와 평균만 낸다. 따라서 훈련 범위를 벗어나면 기존 데이터 근처의 평균에서 크게 벗어나지 못한다. 교재에서 50cm·100cm 놓어 모두 약 1033g으로 예측한 사례가 이를 보여준다.

#### 18번 정답: ②

[해설] 두 입력 모두 훈련 데이터의 오른쪽 바깥에 있으므로, 가장 가까운 k개 이웃이 동일한 샘플 그룹(약 44~45cm 근방)이 된다. 따라서 동일한 평균값이 계산되어 비슷한 예측이 나온다. 이것이 k-최근접 이

웃 회귀의 외삽(extrapolation) 한계이다.

**19번 정답: ②**

[해설] 교재에서 '선형 회귀는 특성과 타깃 사이의 관계를 가장 잘 나타내는 선형 방정식을 찾는다'고 정의하고 있다. 특성이 하나면 직선 방정식이 되고, 여러 개이면 다차원 선형 방정식이 된다.

**20번 정답: ②**

[해설] 선형 회귀는 특성과 타깃 사이의 관계를 하나의 식으로 학습하기 때문에, 훈련 범위 바깥의 입력에도 직선을 연장해서 예측할 수 있다. 반면 k-최근접 이웃은 저장된 샘플의 평균에 묶여 새로운 추세를 만들지 못한다.

**21번 정답: ②**

[해설] LinearRegression은 sklearn.linear\_model 패키지에 있는 선형 회귀 클래스이다. PolynomialRegression이라는 클래스는 사이킷런에 존재하지 않는다.

**22번 정답: ②**

[해설] coef\_ 속성에는 LinearRegression이 학습한 특성의 계수(기울기)가 저장된다. 교재 예제에서 coef\_가 약 [39.01]로 나왔는데, 이는 길이가 1cm 늘어날 때마다 예측 무게가 약 39g 증가한다는 의미이다.

**23번 정답: ②**

[해설] intercept\_ 속성에는 선형 방정식의 절편(y절편)이 저장된다. 교재 예제에서 intercept\_가 약 -709로 나왔는데, 이는 직선이 y축과 만나는 점을 의미한다.

**24번 정답: ②**

[해설] 교재에서 '모델 파라미터는 선형 회귀가 찾은 가중치처럼 머신러닝 모델이 특성에서 학습한 파라미터를 말함'이라고 정의하고 있다. 반면 alpha, n\_neighbors 등 사람이 사전에 지정하는 값은 하이퍼파라미터이다.

**25번 정답: ②**

[해설] 사이킷런에는 PolynomialRegression이라는 클래스가 없다. 다항 특성을 만들어주는 것은 PolynomialFeatures이고, 실제 학습은 LinearRegression이 수행한다. 교재 확인 문제에서도 이 점을 명확히 묻고 있다.

**26번 정답: ②**

[해설] 교재에서 '이 함수는 비선형일 수 있지만 여전히 선형 회귀로 표현할 수 있음'이라고 설명하고 있다. 예를 들어  $ax^2 + bx + c$ 는 길이에 대해서는 비선형이지만, 계수 a, b, c에 대해서는 선형이므로 LinearRegression으로 학습 가능하다.

**27번 정답: ②**

[해설] 교재에서 'column\_stack() 함수 사용: train\_input을 제공한 것과 train\_input 두 배열을 나란히 붙이면 된다'고 설명하고 있다. 이 함수는 열 방향으로 배열을 합쳐 2차원 배열을 만든다.

**28번 정답: ②**

[해설] 교재에서 다중 회귀를 '여러 개의 특성을 사용한 선형 회귀'로 정의하고 있다. 예제에서는 농어의 길이, 높이, 두께 3개 특성을 함께 사용하였다.

**29번 정답: ②**

[해설] 교재에서 특성 공학을 '기존의 특성을 사용해 새로운 특성을 뽑아내는 작업'으로 정의하고 있다. 예를 들어 길이<sup>2</sup>, 길이×높이, 높이<sup>2</sup> 같은 새로운 특성을 기존 길이·높이·두께로부터 만들어내는 것이다.

**30번 정답: ②**

[해설] PolynomialFeatures는 sklearn.preprocessing 패키지에 있는 변환기 클래스로, 입력 특성으로부터 제곱, 교호항 등 다항 특성을 자동으로 만들어준다.

**31번 정답: ②**

[해설] 교재에서 '변환기 클래스는 모두 fit(), transform() 메서드를 제공'한다고 명시하고 있다. 반면 모델 클래스(추정기)는 fit(), score(), predict()를 제공한다.

**32번 정답: ①**

[해설] 교재에서 변환기는 '특성을 만들거나 전처리하기 위한 클래스'이고, 추정기(estimator)는 'LinearRegression 같은 모델 클래스'라고 구분하고 있다. 변환기의 fit()은 입력 데이터만 전달하며 타깃이 필요하지 않다.

**33번 정답: ②**

[해설] 교재에서 '사이킷런의 선형 모델은 자동으로 절편을 추가하므로 굳이 이렇게 특성을 만들 필요가 없다'고 설명하면서, include\_bias=False로 절편용 1 특성을 제외하는 방법을 보여주고 있다.

**34번 정답: ②**

[해설] 교재에서 규제를 '머신러닝 모델이 훈련 세트를 너무 과도하게 학습하지 못하도록 휘방하는 것'이라고 정의하고 있다. 선형 회귀의 경우 특성에 곱해지는 계수의 크기를 작게 만드는 방식으로 적용한다.

**35번 정답: ②**

[해설] 교재 그림에서 왼쪽(규제 없음)은 훈련 세트를 과도하게 학습한 모델이고, 오른쪽(규제 적용)은 기울기를 줄여 보다 보편적인 패턴을 학습한 모델이다. 규제는 계수의 크기를 작게 만들어 이런 효과를 달성한다.

**36번 정답: ②**

[해설] 교재에서 '릿지(ridge): 계수를 제공한 값을 기준으로 규제를 적용'한다고 명시하고 있다. 이를 L2 규제라고도 한다.

**37번 정답: ②**

[해설] 교재에서 '라쏘(lasso): 계수의 절댓값을 기준으로 규제를 적용'한다고 명시하고 있다. 이를 L1 규제라고도 한다.

**38번 정답: ②**

[해설] 교재에서 '라쏘는 계수값을 0으로도 만들 수 있다'고 명시하고 있다. 55개 특성 중 40개의 계수를

0으로 만든 교재 예제가 이를 보여준다. 이런 특징 때문에 라쏘는 유용한 특성을 골라내는 용도로도 사용할 수 있다.

**39번 정답: ③**

[해설] 교재에서 '모델 객체를 만들 때 alpha 매개변수로 규제 강도를 조절'한다고 설명하고 있다. alpha 값이 크면 규제가 세지고, 작으면 규제가 약해져서 일반 선형 회귀에 가까워진다.

**40번 정답: ③**

[해설] 교재에서 'alpha 값이 크면 규제 강도가 세지므로 계수 값을 더 줄이고 조금 더 과소적합되도록 유도'한다고 설명하고 있다. 릿지 alpha 그래프에서도 오른쪽(alpha가 큰 쪽)으로 갈수록 두 점수가 모두 낮아지는 과소적합 양상을 보인다.

**41번 정답: ②**

[해설] 교재에서 'alpha 값이 작으면 계수를 줄이는 역할이 줄어들고 선형 회귀 모델과 유사해지므로 과대적합될 가능성이 큼'이라고 명시하고 있다. alpha 그래프 왼쪽에서 훈련/테스트 점수 차이가 크게 벌어지는 과대적합 양상을 확인할 수 있다.

**42번 정답: ③**

[해설] 교재에서 '특성의 스케일이 정규화되지 않으면 여기에 곱해지는 계수 값도 차이가 나며, 계수 값의 크기가 서로 많이 다르면 공정하게 제어되지 않는다'고 설명하고 있다. 따라서 규제 적용 전에 표준화가 필요하다.

**43번 정답: ②**

[해설] 교재에서 '사이킷런에서 제공하는 StandardScaler 클래스를 사용'하여 표준화한다고 설명하고 있다. StandardScaler의 fit()으로 훈련 세트의 평균과 표준편차를 학습하고, transform()으로 실제 변환을 수행한다.

**44번 정답: ②**

[해설] 교재에서 '모델 파라미터는 머신러닝 모델이 특성에서 학습한 파라미터'(예: coef\_, intercept\_)이고, '하이퍼파라미터는 머신러닝 알고리즘이 학습하지 않는 파라미터로 사람이 사전에 지정해야 한다'(예: alpha, n\_neighbors)고 구분하고 있다.

**45번 정답: ②**

[해설] alpha는 릿지·라쏘의 규제 강도를 설정하는 대표적인 하이퍼파라미터로, 사람이 사전에 지정해야 한다. coef\_와 intercept\_는 모델이 학습하는 모델 파라미터이고, R<sup>2</sup>와 MAE는 평가 지표이다.

**46번 정답: ②**

[해설] 교재에서 'PolynomialFeatures 클래스의 degree 매개변수를 사용하여 필요한 고차항의 최대 차수를 지정'한다고 설명하고 있다. 기본값은 2이며, 교재에서는 degree=5로 설정하여 55개 특성을 만든 사례를 보여준다.

**47번 정답: ②**

[해설] 교재에서 참새 비유를 들어 설명하고 있다. '42개의 참새를 맞추기 위해 한 번만 쓸 수 있다면 중앙

을 겨냥해야 하지만, 55번이나 쓸 수 있다면 한 번에 하나씩 모든 참새를 맞출 수 있다.' 즉 특성이 샘플보다 많으면 모델이 훈련 데이터를 거의 완벽하게 외워 과대적합된다.

**48번 정답: ②**

[해설] 4개의 원소를  $2 \times 2 = 4$ 로 재배열하므로 원소 수가 일치한다. 따라서 shape는 (2, 2)가 된다. 교재 슬라이드 11에서 동일한 예제를 사용하고 있다.

**49번 정답: ②**

[해설] 교재에서 '지정한 크기가 원본 배열에 있는 원소의 개수와 다르면 에러가 발생'한다고 명시하고 있다. 원소는 4개인데  $2 \times 3 = 6$ 개 크기로 바꾸려고 했으므로 에러가 발생한다.

**50번 정답: ③**

[해설] `reshape(-1, 1)`에서 열 수를 1로 고정하고, -1은 행 수를 자동 계산한다. 원소 3개 ÷ 열 1개 = 행 3개이므로 shape는 (3, 1)이 된다. 교재에서 `train_input.reshape(-1, 1)`로 (42, 1) shape를 만든 것과 동일한 원리이다.

**51번 정답: ②**

[해설] 교재에서 k-최근접 이웃 회귀로 50cm와 100cm 농어를 예측했을 때 모두 약 1033g이 나온 사례를 보여준다. 이는 훈련 범위를 벗어나면 가장 가까운 같은 이웃 그룹의 평균만 반환하는 k-최근접 이웃 회귀의 구조적 한계 때문이다.

**52번 정답: ②**

[해설] `coef_`는 기울기(약 39.01), `intercept_`는 절편(약 -709.02)을 나타낸다. 즉 학습된 방정식은 '무게  $\approx 39.01 \times$  길이 - 709.02'이다. 교재 슬라이드 29에서 이 값을 확인하고 있다.

**53번 정답: ①**

[해설] 예측 무게 =  $39.01 \times 30 + (-709.02) = 1170.3 - 709.02 = 461.28 \approx$  약 462g이다. 선형 방정식  $y = ax + b$ 에  $x=30$ 을 대입한 결과이다.

**54번 정답: ②**

[해설] 교재에서 '직선 모델은 단순하여 농어의 무게가 음수일 수도 있다'고 지적하고 있다.  $y = 39.01 \times 5 - 709.02 = 195.05 - 709.02 = -513.97$ 로 음수가 나온다. 이것이 단순 직선 모델의 한계이며, 다항 회귀가 필요한 이유이다.

**55번 정답: ②**

[해설] `column_stack`은 열 방향으로 배열을 붙인다. `train_input2(42,1)`과 `train_input(42,1)`을 나란히 붙이면 (42, 2)가 된다. 교재에서도 '원래 특성인 길이를 제공하여 왼쪽 열에 추가했기 때문에 열이 2개로 늘어남'이라고 설명하고 있다.

**56번 정답: ②**

[해설] `train_poly`가 [길이<sup>2</sup>, 길이] 순서로 구성되었으므로, 첫 번째 계수 1.01은 길이<sup>2</sup>에, 두 번째 계수 -21.6은 길이에 곱해진다. 교재 슬라이드 33에서 '무게 =  $1.01 \times$  길이<sup>2</sup> - 21.6  $\times$  길이 + 116.05'라고 명시하고 있다.

**57번 정답: ③**

[해설] 교재에서 '판다스의 read\_csv() 함수에 주소를 삽입'하면 데이터프레임이 만들어진다고 설명하고 있다. 데이터프레임은 판다스의 핵심 자료구조로, 행과 열을 가진 표 형태의 데이터이다.

**58번 정답: ③**

[해설] 원래 특성 3개(길이, 높이, 두께) + 제곱항 3개(길이<sup>2</sup>, 높이<sup>2</sup>, 두께<sup>2</sup>) + 교호항 3개(길이×높이, 길이×두께, 높이×두께) = 총 9개이다. 교재에서 train\_poly.shape가 (42, 9)로 나온 것이 이를 확인해준다.

**59번 정답: ③**

[해설] 교재에서 '5제곱까지 특성을 만들어 출력'한 결과 train\_poly.shape가 (42, 55)로 나온다고 명시하고 있다. 3개 특성의 5차 다항식 조합(bias 제외)은 총 55개의 특성을 생성한다.

**60번 정답: ③**

[해설] 훈련 점수는 거의 1(완벽)이지만 테스트 점수는 -144(매우 큰 음수)이므로 심각한 과대적합 상태이다. 교재에서 55개 특성으로 42개 샘플을 훈련한 결과로 이 수치를 보여주며, '훈련 세트에 너무 과대적합되므로 테스트 세트에서는 형편없는 점수를 만들게 됨'이라고 설명하고 있다.

**61번 정답: ②**

[해설] 교재에서 '꼭 훈련 세트로 학습한 변환기를 사용해 테스트 세트까지 변환해야 함'이라고 강조하고 있다. 훈련 세트의 평균과 표준편차를 기준으로 테스트 세트도 동일하게 변환해야 데이터가 같은 스케일에 놓이고, 공정한 평가가 가능하다.

**62번 정답: ③**

[해설] include\_bias=False이므로 절편용 1이 제외된다. 원래 특성(2, 3), 각각의 제곱(4=2<sup>2</sup>, 9=3<sup>2</sup>), 교호항(6=2×3)이 생성되어 [2. 3. 4. 6. 9.]가 된다. 교재 슬라이드 44에서 동일한 예제를 보여주고 있다.

**63번 정답: ②**

[해설] 교재에서 alpha를 0.001부터 100까지 10배씩 늘려가며 훈련/테스트 점수를 그래프로 그리고, '두 그래프가 가장 가깝고 테스트 세트의 점수가 가장 높은' 지점을 최적 alpha로 선택하고 있다.

**64번 정답: ②**

[해설] 교재에서 'alpha 값을 0.001부터 10배씩 늘렸기 때문에 이대로 그래프를 그리면 그래프 왼쪽이 너무 촘촘해진다. alpha\_list에 있는 6개의 값을 동일한 간격으로 나타내기 위해서는 x축을 로그 스케일로 나타내야 한다'고 설명하고 있다.

**65번 정답: ②**

[해설] lasso.coef\_ == 0은 각 계수가 0인지를 True/False로 확인하고, np.sum()은 True의 개수를 센다. 교재에서 이 결과가 40이므로 '55개의 특성을 모델에 주입했지만 라쏘 모델이 사용한 특성은 15개에 불과하다'고 해석하고 있다.

**66번 정답: ③**

[해설] 55개 전체 특성에서 계수가 0인 40개를 빼면 실제로 사용되는 특성은 15개이다. 교재에서 '이런 특징 때문에 라쏘 모델을 유용한 특성을 골라내는 용도로도 사용할 수 있다'고 설명하고 있다.

**67번 정답: ②**

[해설] 교재에서 '사이킷런의 라쏘 모델은 최적의 계수를 찾기 위해 반복적인 계산을 수행하는데, 지정한 반복 횟수가 부족할 때 이런 경고가 발생'한다고 설명하고 있다.

**68번 정답: ③**

[해설] 교재에서 '이 반복 횟수를 충분히 늘리기 위해 max\_iter 매개변수의 값을 10000으로 지정했다'고 설명하고 있다. max\_iter는 알고리즘의 수행 반복 횟수를 지정하며, 기본값은 1000이다.

**69번 정답: ④**

[해설] evaluate()는 딥러닝 라이브러리인 케라스에서 사용하는 메서드이며, 사이킷런의 기본 모델 클래스에는 없다. 사이킷런의 기본 패턴은 fit(), predict(), score()이다. 교재 확인 문제(슬라이드 38)에서도 동일한 문제를 다루고 있다.

**70번 정답: ②**

[해설] 일반적으로 훈련 세트 점수가 테스트 세트보다 약간 높아야 자연스럽다. 테스트 점수(0.98)가 훈련 점수(0.97)보다 높으므로 과소적합 경향이 있다. 교재에서 k=5일 때 비슷한 양상이 나타나며, k를 3으로 줄여 해결한 사례를 보여준다.

**71번 정답: ③**

[해설] 교재에서 '사이킷런의 k-최근접 이웃 알고리즘의 기본 k 값은 5'라고 명시하고 있다. 교재 예제에서 이 기본값 5에서 과소적합이 나타나 3으로 줄인 사례를 다룬다.

**72번 정답: ③**

[해설] 교재에서 'kneighbors() 메서드를 사용하여 50cm 농어의 이웃을 구한다'고 설명하며, distances(거리)와 indexes(인덱스) 두 값을 반환받고 있다. 이를 통해 어떤 훈련 샘플이 이웃으로 선택되었는지 확인할 수 있다.

**73번 정답: ②**

[해설] 교재 코드에서 mean\_absolute\_error(test\_target, test\_prediction) 순서로 호출하고 있다. 첫 번째 인수에 실제 타깃값, 두 번째 인수에 모델의 예측값을 전달한다.

**74번 정답: ④**

[해설]  $a \times b^3$ 의 총 차수는  $1+3=4$ 로, degree=3(최고 차수 3)을 초과하므로 생성되지 않는다. 교재 확인 문제(슬라이드 61)에서 동일한 유형의 문제를 다루고 있다. 1,  $a \times b$ (차수2),  $a^2 \times b$ (차수3),  $a^3$ (차수3)은 모두 포함된다.

**75번 정답: ③**

[해설] 교재에서 Ridge 클래스의 alpha 기본값은 1이라고 설명하고 있다. 교재 예제에서는 alpha=0.1이 최적으로 나왔다.

**76번 정답: ②**

[해설] 교재에서 '이 모델은 훈련 세트와 테스트 세트의 점수가 비슷하게 모두 높고 과대적합과 과소적합 사이에서 균형을 맞추었다'고 설명하고 있다. 두 점수가 모두 0.98~0.99 수준으로 높으면서 차이가 크지

않다.

**77번 정답: ③**

[해설] 교재에서 산점도를 보고 '점들이 오른쪽 위로 갈수록 올라가는 경향이 보인다. 즉 길이가 길수록 무게가 커지는 양의 상관관계가 있다'고 설명하고 있다.

**78번 정답: ②**

[해설] 교재에서 '머신러닝에서 종종 가중치는 방정식의 기울기와 절편을 모두 의미하는 경우가 많다'고 설명하고 있다. 즉 가중치는 모델이 학습한 모든 파라미터(계수 + 절편)를 포괄하는 용어이다.

**79번 정답: ②**

[해설] 교재에서 단순 직선 모델의 훈련 점수가 0.94로 과소적합이었는데, 길이<sup>2</sup>를 추가하여 다항 회귀를 적용하자 훈련 점수 0.97, 테스트 점수 0.98로 개선되었다. 곡선 형태가 데이터의 실제 구조를 더 잘 반영했기 때문이다.

**80번 정답: ②**

[해설] 교재에서 '55개의 특성을 모델에 주입했지만 라쏘 모델이 사용한 특성은 15개에 불과하며, 이런 특징 때문에 라쏘 모델을 유용한 특성을 골라내는 용도로도 사용할 수 있다'고 설명하고 있다.

**81번 정답: ②**

[해설] 교재에서 k-최근접 이웃 회귀는 훈련 범위 안(보간)에서는 괜찮지만, 범위 밖(외삽)에서는 새로운 추세를 만들어 내지 못한다고 설명하고 있다. 선형 회귀는 식을 학습하므로 외삽이 가능하다.

**82번 정답: ②**

[해설] 교재 마무리(슬라이드 59)에서 read\_csv()의 주요 매개변수를 정리하며, 'sep은 CSV 파일의 구분자를 지정하고 기본값은 콤마(,)이다'라고 설명하고 있다.

**83번 정답: ③**

[해설] 3개 특성에 degree=2, include\_bias=False를 적용하면: 원래 3개 + 제곱 3개( $a^2, b^2, c^2$ ) + 교호항 3개( $ab, ac, bc$ ) = 총 9개 특성이 생성된다. 교재에서 train\_poly.shape가 (42, 9)로 나온 것과 일치한다.

**84번 정답: ②**

[해설] 교재에서 'get\_feature\_names\_out() 메서드를 호출하여 9개의 특성이 각각 어떤 입력의 조합으로 만들어졌는지 알려준다'고 설명하고 있다. 출력 예시로 'length', 'length^2', 'length height' 등이 나온다.

**85번 정답: ③**

[해설] 교재에서 '훈련 세트에서 학습한 평균과 표준편차는 StandardScaler 클래스 객체의 mean\_, scale\_ 속성에 저장된다. 특성마다 계산하므로 55개의 평균과 표준편차가 들어 있다'고 설명하고 있다.

**86번 정답: ②**

[해설] alpha=0.001은 훈련/테스트 차이가 크므로 과대적합, alpha=100은 두 점수 모두 낮아 과소적합이다. alpha=0.1은 두 점수가 모두 높으면서 차이도 작으므로 과대·과소적합의 균형이 가장 잘 잡혀 있다.

**87번 정답: ③**

[해설] 교재에서 '특성이 2개면 타깃값과 함께 3차원 공간을 형성하고 선형 회귀 방정식은 평면이 된다'고 설명하고 있다. 특성이 1개면 직선, 2개면 평면, 3개 이상이면 초평면이 된다.

**88번 정답: ②**

[해설] 교재 마무리(슬라이드 60)에서 '이 클래스는 최적의 모델을 찾기 위해 좌표축을 따라 최적화를 수행해가는 좌표 하강법(coordinate descent)을 사용'한다고 명시하고 있다.

**89번 정답: ③**

[해설] 교재 마무리(슬라이드 60)에서 'solver 매개변수에 최적의 모델을 찾기 위한 방법을 지정할 수 있으며, 기본값은 auto이고 데이터에 따라 자동으로 선택'된다고 설명하고 있다.

**90번 정답: ②**

[해설] 교재 마무리(슬라이드 60)에서 'interaction\_only가 True이면 거듭제곱 항은 제외되고 특성 간의 곱셈 항만 추가된다'고 설명하고 있다. 즉  $a^2$ ,  $b^2$  같은 제곱항은 빠지고,  $a \times b$  같은 교호항만 남는다.

**91번 정답: ①**

[해설] 사이킷런 모델 사용 순서는 항상 ① 객체 생성(다) → ② 훈련(나: fit) → ③ 예측(가: predict)이다. 교재의 모든 코드 예제에서 이 순서를 일관되게 따르고 있다.

**92번 정답: ②**

[해설] 교재에서  $R^2$ 로 모델의 설명력을 확인한 뒤, MAE로 '예측이 평균적으로 19g 정도 차이가 난다'는 구체적 오차를 확인하고 있다.  $R^2$ 는 비율, MAE는 단위가 있는 오차이므로 둘을 함께 보면 모델 성능을 더 입체적으로 이해할 수 있다.

**93번 정답: ②**

[해설] 교재에서 '릿지와 라쏘 모두 sklearn.linear\_model 패키지에 있다고 명시하고 있다. LinearRegression과 같은 패키지이다.

**94번 정답: ③**

[해설] 스팸 여부는 '스팸/정상' 두 클래스 중 하나를 예측하는 분류 문제이다. 교재에서 회귀를 '임의의 수치를 예측하는 문제'로 정의하고, 분류는 '클래스 중 하나로 분류하는 것'으로 구분하고 있다.

**95번 정답: ②**

[해설] 교재에서 '테스트 세트의 점수는 분류에서는 정확도, 회귀에서는 결정계수( $R^2$ )'라고 명시하고 있다. 즉 같은 score() 메서드라도 분류 모델은 정확도를, 회귀 모델은  $R^2$ 를 반환한다.

**96번 정답: ②**

[해설] 과대적합은 훈련 세트에서 점수가 굉장히 좋지만 테스트 세트에서는 점수가 나쁜 상태이다. '테스트 세트의 점수도 높다'는 것은 과대적합이 아니라 정상적인 모델에 가깝다. 교재 확인 문제(슬라이드 62)에서도 동일한 유형을 다루고 있다.

**97번 정답: ③**

[해설] 교재에서 '이웃의 개수를 3으로 설정합니다'라는 주석과 함께 `knr.n_neighbors = 3` 코드를 보여주고 있다. 새 객체를 만들지 않고 기존 객체의 속성을 직접 변경하여 하이퍼파라미터를 바꿀 수 있다.

**98번 정답: ③**

[해설] 교재에서 50cm 농어의 최근접 이웃을 시각화할 때, `train_input[indexes]`는 이웃으로 선택된 샘플들이므로 다이아몬드(D) 마커로, 예측 대상인 50cm 농어는 삼각형(^) 마커로 표시하고 있다. 이를 통해 어떤 훈련 샘플이 예측에 영향을 미쳤는지 시각적으로 확인할 수 있다.

**99번 정답: ②**

[해설] 교재에서 '라쏘 모델을 훈련할 때 `ConvergenceWarning`이 발생할 수 있으며, 반복 횟수를 충분히 늘리기 위해 `max_iter` 매개변수의 값을 10000으로 지정했다'고 설명하고 있다. 기본값 1000으로는 최적 계수를 찾기에 반복이 부족할 수 있다.

**100번 정답: ②**

[해설] `train_poly`를 만들 때 `np.column_stack((train_input**2, train_input))`으로 (길이<sup>2</sup>, 길이) 순서의 2차원 배열을 구성했다. 모델은 이 구조로 훈련되었으므로 예측할 때도 동일하게 [50<sup>2</sup>=2500, 50] 형태로 넣어야 한다. 훈련 데이터와 예측 데이터의 특성 구조가 일치해야 올바른 예측이 가능하다.