<table>
<tr><th>위치</th><th>오류유형</th><th>수정 전</th><th>수정 후</th></tr>
<tr><td>16p</td><td>문제-본문</td><td>
<table>
<tr><th>키워드</th><th>사용법</th><th>설 명</th></tr>
<tr><td>import</td><td>import A, B, ...</td><td>· 모듈 또는 함수를 호출한다.
· 콤마를 사용하여 여러 개를 동시에 호출할 수 있다.</td></tr>
<tr><td>from</td><td>from A, a import ab</td><td>· 모듈의 일부 함수 또는 패키지만 호출할 수 있다.
· 구두점(.)을 활용하여 함수의 위치를 구체적으로 지정할 수 있다.</td></tr>
<tr><td>as</td><td>import ABCD as ad</td><td>· 호출하는 대상의 약칭을 설정한다.</td></tr>
</table>
</td><td>
<table>
<tr><th>키워드</th><th>사용법</th><th>설 명</th></tr>
<tr><td>import</td><td>import pandas, numpy</td><td>· 모듈 또는 함수를 호출한다.
· 콤마를 사용하여 여러 개를 동시에 호출할 수 있다.</td></tr>
<tr><td>from</td><td>from pandas import DataFrame
from sklearn.datasets import load_iris</td><td>· 모듈의 일부 함수 또는 패키지만 호출할 수 있다.
· 구두점(.)을 활용하여 함수의 위치를 구체적으로 지정할 수 있다.</td></tr>
<tr><td>as</td><td>import numpy as np</td><td>· 호출하는 대상의 약칭을 설정한다.</td></tr>
</table>
</td></tr>
<tr><td>17p</td><td>문제-본문</td><td>다음 예제에서 2~3번째 줄과 4번째 줄, 5번째 줄은 모두 같은 의미이다.
In [3]:

```
import numpy as np
dataset = np.array([['kor', 70], ['math', 80]])
df = pd.DataFrame(dataset, columns=['class', 'score'])
df = pd.DataFrame(data=[['kor', 70], ['math', 80]], columns=['class', 'score'])
df = pd.DataFrame( {'class': ['kor', 'math'], 'score': [70, 80]})
df
```

</td><td>삭제
In [3]:

```
import numpy as np
dataset = np.array([['kor', 70], ['math']])
df = pd.DataFrame(dataset, columns=['class', 'score'])
df
```

</td></tr>
<tr><td>17p</td><td>문제-본문</td><td colspan="2">
수정 전
Tip
Series 선언하기
pd.Series({'idx 1': value, 'idx 2': value}, name='class'

수정 후
Tip
Series 선언하기
pd.Series({'idx 1': value, 'idx 2': value}, name='class'
In [3]에서는 2번째 줄에서 array를 선언하고 3번째 줄에서 array를 데이터프레임으로 만들고 컬럼명을 추가하여 Out [3]의 데이터프레임을 선언하였다. 데이터프레임을 선언하는 것은 이 방법 외에도 다양한 방법으로 수행할 수 있다. 아래의 두 줄은 각각 Out [3]과 같은 데이터프레임을 한 줄의 코드로 선언하는 다른 방법이다.

```
df = pd.DataFrame([['kor', 70], ['math']], columns=['class', 'score'])
df = pd.DataFrame({'class': ['kor', 'math'], 'score': [70, 80]})
```

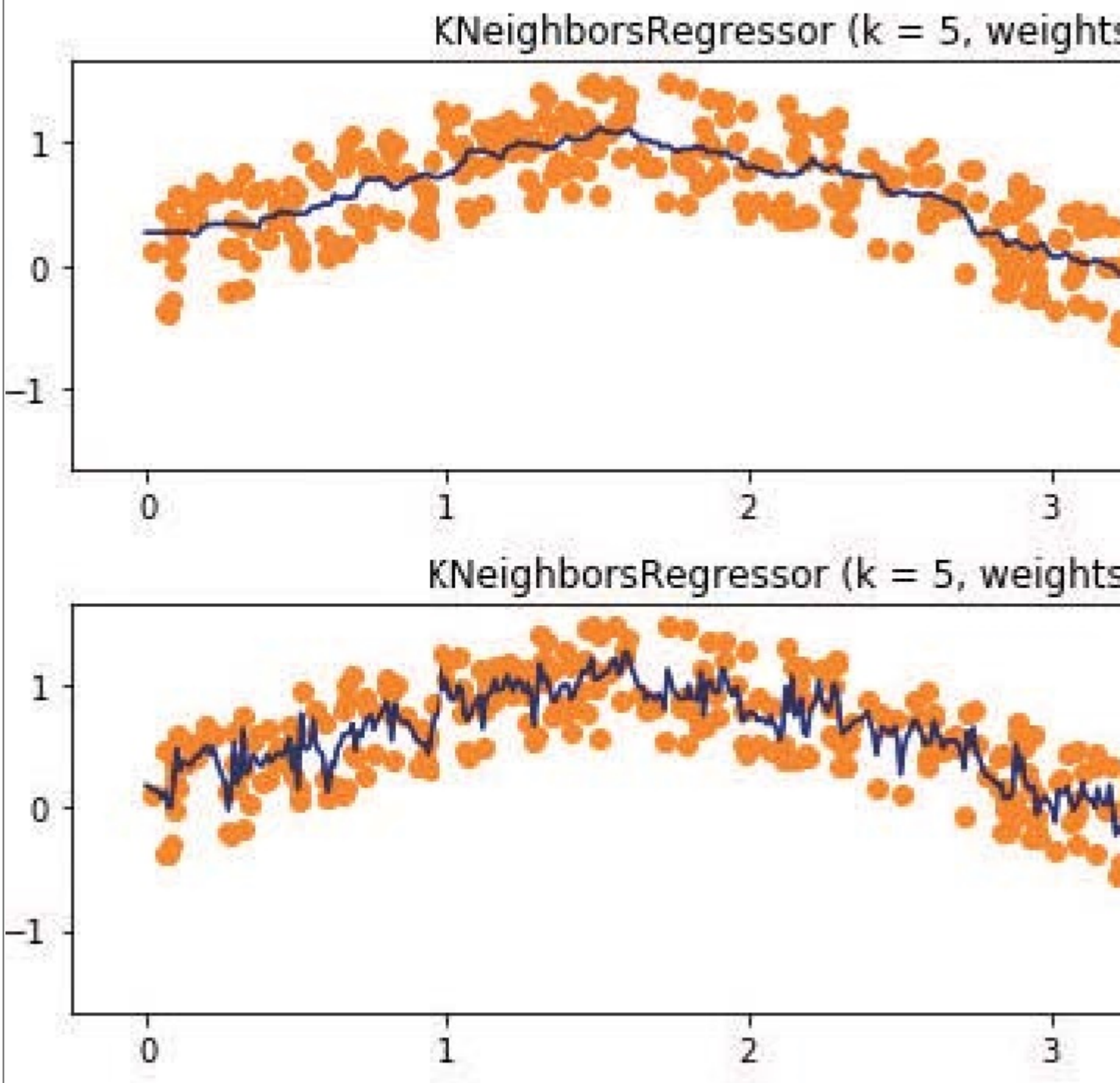
</td></tr>
<tr><td>153p</td><td>문제-본문</td><td>1. 개 념
선형 회귀는 입력 특성에 대한 선형함수를 만들어 예측을 하는 알고리즘이다. 독립변수가 하나인 경우 특정 직선을 학습하는 것이다.</td><td>1. 개 념
선형 회귀는 입력 특성에 대한 선형함수를 만들어 예측을 하는 알고리즘이다. 독립변수가 하나인 경우 데이터의 특징을 가장 잘 설명하는 직선을 학습하는 것이다.</td></tr>
<tr><td>194p</td><td>문제-본문</td><td>predic_proba 메서드d와 decision_function 메서드를 사용해 로지스틱 회귀 그래프를 그려볼 수 있다.</td><td>predict_proba 메서드와 decision_function 메서드를 사용해 로지스틱 회귀 그래프를 그려볼 수 있다.</td></tr>
</table>

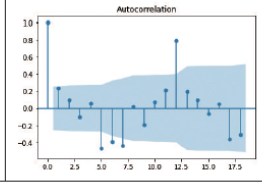
위치	오류유형	수정 전	수정 후
230p	문제-본문		

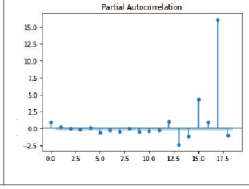
In [6]:

```
(생략)
plt. title("KNeighborsRegressor (k = %i, weights = '%s')" % (5, weights))
plt. tight_layout()
plt. show()
```

Out [6]: 아래 그래프 참조

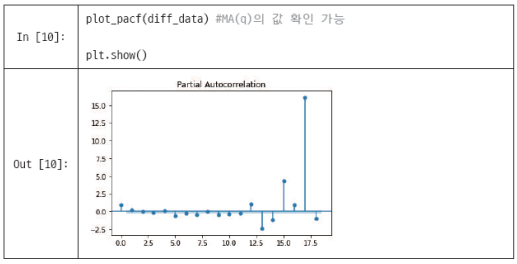
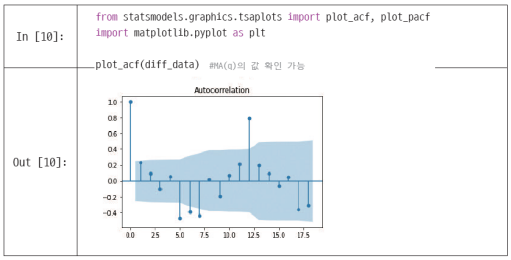


<table>
<tr><th>위치</th><th>오류유형</th><th>수정 전</th><th>수정 후</th></tr>
<tr><td>249p</td><td>문제-본문</td><td>BaggingClassifier를 사용해 분류기를 생성하여 예측을 수행해보자.
In [5]:
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
clf = BaggingClassifier(base_estimator = DecisionTreeClassifier())
pred = clf.fit(x_train, y_train).predict(x_test)
print("Accuracy Score : ", clf.score(x_test, pred))
Out [5]:
Accuracy Score : 1.0</td><td>BaggingClassifier를 사용해 분류기를 생성하여 예측을 수행해보자.
In [5]:
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
clf = BaggingClassifier(base_estimator = DecisionTreeClassifier())
pred = clf.fit(x_train, y_train).predict(x_test)
print("Accuracy Score : ", clf.score(x_test, y_test))
Out [5]:
Accuracy Score : 0.9122807017543859</td></tr>
<tr><td>253p</td><td>문제-본문</td><td>훈련 단계에서 알고리즘은 각 모델에 가중치를 할당하므로 분류결과가 좋은 데이터는 높은 가중치를, 분류결과가 좋지 않은 데이터는 낮은 가중치를 할당받아 다음 붓스트래핑에서 추출될 확률이 높아진다. 따라서 배깅에 비해 모델의 장점을 최적화하고 train 데이터에 대해 오류가 적은 결합모델을 생성할 수 있다는 장점이 있다.</td><td>훈련 단계에서 알고리즘은 데이터 샘플에 가중치를 할당하므로 분류결과가 좋지 않은 데이터는 높은 가중치를, 분류결과가 좋은 데이터는 낮은 가중치를 할당받는다. 높은 가중치를 받은 데이터 샘플은 다음 붓스트래핑에서 추출될 확률이 높아진다. 직전 단계에서 예측력이 약했던 부분을 다음 단계에서 개선해 나간다는 개념인 것이다. 따라서 배깅에 비해 모델의 장점을 최적화하고 train 데이터에 대해 오류가 적은 결합모델을 생성할 수 있다는 장점이 있다.</td></tr>
<tr><td>371p</td><td>문제-본문</td><td>② ACF(Auto Correlation Function)
ACF는 자기상관 함수로 이 값은 시차에 따른 자기상관성을 의미한다. ACF 값을 시차에 따른 그래프로 시각화를 해보면, 최적의 p 값을 찾을 수 있다. 비정상 시계열일 경우에는 ACF 값은 느리게 0에 접근하며, 양수의 값을 가질 수도 있다. 하지만 정상 시계열일 경우에는 ACF 값이 빠르게 0으로 수렴하며, 0으로 수렴할 때에 시차를 p 값으로 설정한다.
(생략)
③ 파이썬을 활용한 AR 모형의 p 값 찾기
In [9]:
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt
plot_acf(diff_data) #AR(p)의 값 확인 가능
Out [9]:
 

ACF 값을 확인해 보았을 때, 시차 2 이후에 0에 수렴하는 것을 알 수 있다. 그러므로 AR 모형에서 최적의 p 값은 2로 설정할 수 있다.</td><td>② PACF(Partial Auto Correlation Function)
PACF는 편자기상관 함수이다. PACF는 ACF와는 다르게 시차가 다른 두 시계열 데이터 간의 순수한 상호 연관성을 나타낸다. 그러므로 PACF 값이 0에 수렴할 때의 p값을 AR 모형의 p값으로 설정한다.
③ 파이썬을 활용한 AR 모형의 p 값 찾기
In [9]:
plot_pacf(diff_data) #AR(p)의 값 확인 가능
plt.show()
Out [9]:
 

제7장 통계분석 371
PACF 값을 확인해 보았을 때, 시차 2 이후에 0에 수렴하는 것을 알 수 있다. 그러므로 AR 모형에서 최적의 p 값은 2로 설정할 수 있다.</td></tr>
</table>

위치	오류유형	수정 전	수정 후

372p	문제-본문	② PACF(Partial Auto Correlation Function)
		PACF는 편자기상관 함수이다. PACF는 ACF와는 다르게 시차가 다른 두 시계열 데이터 간의 순수한 상호 연관성을 나타낸다. 그러므로 PACF 값이 0에 수렴할 때의 q 값을 MA 모형의 q 값으로 설정한다.
		(생략)
		③ 파이썬을 활용한 AR 모형의 q 값 찾기
		 
		In [10]:
		plot_pacf(diff_data) #MA(q)의 값 확인 가능
		plt.show()
		Out [10]:
		Partial Autocorrelation
		PACF 값을 확인해 보았을 때, 시차 2 이후에 0에 수렴하는 것을 알 수 있다. 그러므로 MA 모형에서 최적의 q 값은 2로 설정할 수 있다.
			② ACF(Auto Correlation Function)
			ACF는 자기상관 함수로 이 값은 시차에 따른 자기상관성을 의미한다. ACF 값을 시차에 따른 그래프로 시각화를 해보면, 최적의 q값을 찾을 수 있다. 비정상 시계열일 경우에는 ACF 값은 느리게 0에 접근하며, 양수의 값을 가질 수도 있다. 하지만 정상 시계열일 경우에는 ACF 값이 빠르게 0으로 수렴하며, 0으로 수렴할 때에 시차를 q로 설정한다.
			③ 파이썬을 활용한 ~~AR~~ MA 모형의 q 값 찾기
			 
			In [10]:
			from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
			import matplotlib.pyplot as plt
			plot_acf(diff_data) #MA(q)의 값 확인 가능
			Out [10]:
			Autocorrelation
			372 파이썬 한권으로 끝내기
			ACF 값을 확인해 보았을 때, 시차 2 이후에 0에 수렴하는 것을 알 수 있다. 그러므로 MA 모형에서 최적의 q값은 2로 설정할 수 있다.

387p	문제-본문	종속변수 학생 성적에 대한 분포를 확인한 결과 평균 근처에 관측치가 많은 정규분포 형태를 띄었다. Shapriro test 결과도 p-value가 0.05 이하로 정규성을 띄고 있음을 확인할 수 있었다. 또한 해당 종속변수의 값이 정규성을 띄고 다양한 값이 있으므로 다중 classification으로 예측하는 것보다는 회귀분석을 진행하는 것이 좋을 것으로 판단된다.	종속변수 학생 성적에 대한 분포를 확인한 결과 평균 근처에 관측치가 많아 정규분포 형태로 보이나, Shapiro test 결과 p-value의 값이 0.05 이하로 정규분포를 따른다고 할 수는 없다. 하지만 종속변수의 값이 종 모양을 갖추고 있고 다양한 관측치가 존재하므로, 다중 classification 보다는 회귀분석으로 예측하는 것이 좋다고 판단된다.
388p	문제-본문	③ 종속변수 분포 설명 종속변수의 분포는 정규분포를 띈다.	③ 내용 삭제
404p	문제-본문	3. ANOVA 분석 변수 3개(하나는 범주형 변수/나머지 두 개는 수치형 연속변수)의 이원분산분석을 수행하고 통계표를 작성하시오.	3. ANOVA 분석 변수 3개(하나는 연속형 변수/나머지 두 개는 범주형 변수)의 이원분산분석을 수행하고 통계표를 작성하시오.

도서의 오류로 학습에 불편드린 점 진심으로 사과드립니다.
더 나은 도서를 만들기 위해 노력하는 시대교육그룹이 되겠습니다.