|  |  |
| --- | --- |
| 교육 제목 | 머신 러닝 |
| 교육 일시 | 2021년 10월 18일 |
| 교육 장소 | 비대면 |
| **교육 내용** | |
| 오전 | \* 어제 다룬내용 복습  \* 선형 회귀 그래프 그리기 sklearn.linear\_model 패키지 아래 linearRegression 클래스로 선형 회귀 알고리즘을 구현하고 훈련메서드 fit() 와 평가메서드 score(), 그리고에측메서드 predict() 를 써서 그래프는 그립니다.  \* 다항 회귀를 그릴려면 2차방정식을 사용해야 하므로 column\_stack 함수를 사용해서 train\_input 과 test\_input 을 제곱해서만들면 됩니다. 원래 특성인 길이를 제곱하여왼쪽열에 추가헀기때문에 훈련세트와 테스트 세트모두 열이 2개로 늘어났습니다 이런방정식을 다항회귀라고 부르며 다항식을 사용한 선형회귀를 다항회귀라고 부릅니다. |
| 오후 | \* 다중 회귀 하나의 특성을 사용한 선형회귀 모델을 훈련해 왔는데, 여러개의 특성을 사용한 선형 회귀를 다중회귀라고 부릅니다. 1개의 특성을 사용했을떄는 직선이지만 2개의특성을 활용하게되면 선형 회귀는 평면을 학습합니다. 특성이 2개면 타긱값과 함께 3차원 공간을 형성하고 선형 회귀 방정식 '타깃 = a\* 특성1 + b \* 특성2 + 절편'은평면이 됩니다. 특성3개이상은 3차원 공간이상을 그리거나 상상하기힘듭니다.  \* 넘파이는 인터넷에서 데이터를 바로 다운로드 하여 사용하는것을 잘지원하지않지만판다스를 사용하면 쉽습니다. 판다스로 자료를 받아서 to\_numpy() 메서드를 사용해 넘파이 배열로 바꿉니다. 사이킷런은 특성을 만들거나 전처리하기 위한 다양한 클래스를 제공하는데 이런 클래스를 변환기라고 부릅니다. 사이킷런의 모델클래스에 일관된 fit(), score(), predict() 메서드가 있는것처럼 변환기 클래스는 fit(), transform() 메서드를 제공합니다. 우리가 사용할 변환기는 polynomialFeatures입니다. 이 클래스는 sklearn.preprocessing 패키지에 포함되어 있습니다. 이 클래스는 기본적으로 각 특성을 제공한 항을 추가하고 특성끼리 서로 곱한 항을 추가합니다. . get\_feature\_names() 메서드를 호출하면 가가 어떤 입력의 조합으로 만들어졌는지 알려줍니다.  \*다중 회귀 모델 훈련하려면 먼저 사이킷런의 LinearRegression 클래스를 임포트하고 앞에서 만든 train\_poly를 사용해 모델을 훈련시켜 보곘습니다. 특성이 늘어나면 선형회귀의 능력은 매우 강하다는것을 알수있는데 , 만약에 3제곱 4제곱 항을 넣으면 어떻게될지 polynomialFeatures 클래스의 degree 매개변수를 사용하여 필요한 고차항의 최대치수를 지정할 수있는데 5제곱까지 특성을 만들어서 출력해 보면, 만들어진 특성의 개가 무려 55개가 되며 이 데이터를 사용해 선형 회귀 모델을 다시 훈련해보면  훈련세트값은 완벽한 점수가 나오지만 테스트 세트의 값은 음수가 나오게됩니다. 이렇게 과대적합을 줄일려면 특성을 줄여야되지만 다른방법으로는 규제를 사용하여 과도하게학습하지 못하도록 훼방하는 것을 말합니다. 즉 훈현 세트에 과대적합되지 않도록 만드는 것이죠. sklearn.preprocessing 의 StandardScaler 클래스를 사용해 클래스의 객체 ss를 초기화한 후 ploynomialFeatures 클래스로 만든 train\_poly를 사용해 이객체를 훈련시킵니다. test\_poly까지 변환시킨후 표준점수로 변환된 train\_scaled와 test\_scaled가준비되면 선형 회귀 모델에 규제를 추가한 모델을 릿지와 라쏘라고 부르는데 각각의두 모델은 규제를 가하는 방법이 다른데 릿지는 계수를 제곱한 값을 기준으로 규제를 적용하고, 라쏘는 계수의 절댓값을 기준으로 규제를 적용합니다.. 일반적으로 릿지를더 선호합니다. 릿지와 라쏘 모두 sklearn.linear\_model 패키지 안에 있습니다.  ridge로 모델객체를 만들고 fit()메서드에서 훈련한 다음 score() 메서드로 평가해보면 거의완벽에 가까웠던 점수가 조금낮아졌습니다. 테스트점수도 정상으로 돌아왔습니다. 릿지와 라쏘 모델을 사용 할때 규제의 양을 임의로 조절할수 있는데 alpha매개변수로 규제의 강도를 조절합니다. 적절한 값을 찾는 한가지 방법은 alpha 값에 대한 R2값의 그래프를 그려 보는 것입니다. 맷플룻립을 임포트하고 alpha 값을 바꿀 때마다 score() 메서드의 결과를 저장할 리스트를 만듭니다. alpha 값을 0.001에서 100까지 10배씩 늘려가며 릿지 회귀 모델을 훈련한 다음 훈련 세트와 메스트 세트의 점수를 파이썬 리스트에 저장합니다 그리고 alpha\_list 에있는 0.001부터 100까지 있는 값을 동일한 간격으로 나타내기위해 로그함수로 바꾸어 지수로 표현하게되면 그래프의 왼쪽을 보면 훈련세트와 케스트 세트의 점수차이가 아주큽니다. 훈련 세트에는 잘 맞고 테스트 세트에는 형편없는 과대적합의 전형적인 모습입니다. 반대로 오른쪽은 훈련 세트와 테스트 세트의 점수가 모두 낮아지는 과소적합으로 가는 모습을 보입니다. |