5주차 예습과제
[5단원]

[5.1] 회귀 소개

회귀분석 :
데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향을 이용한 통계학 기법.
통계학 용어로 여러 개의 독립변수와 한 개의 종속 변수 간의 상관 관계를 모델링하는 기법을 통칭함.
머신러닝 관점에서 보면, 독립 변수는 피처에 해당되며 종속 변수는 결정값임.
머신 러닝 회귀 예측의 핵심은 주어진 피처와 결정 값 데이터 기반에서 학습을 통해 최적의 회귀 계수를 찾아내는 것임.
회귀는 선형/비선형 여부, 독립변수의 개수, 종속변수의 개수에 따라 여러 가지 유형으로 나눌 수 있음. 회귀에서 가장 중요한 것은 바로 회귀 계수임. (단일 회귀, 다중 회귀, 선형 회귀, 비선형 회귀)

지독학습은 두가지 유형으로 나뉨 -> 분류와 회귀.

선형 회귀 : 실제 값과 예측값의 차이를 최소화하는 직선형 회귀선을 최적화 하는 방식. 규제 방법(일반적인 선형 회귀의 과적합 문제를 해결하기 위해서 회귀 계수에 페널티 값을 적용하는 것을 말함)에 따라 다시 별도의 유형으로 나뉠 수 있음.

[5.2] 단순 선형 회귀를 통한 회귀 이해

단순 선형 회귀 : 독립 변수도 하나, 종속변수도 하나인 선형 회귀.
잔차 : 실제 값과 회귀 모델의 차이에 따른 오류 값을 남은 오류, 즉 잔차라고 부름.

최적의 회귀 모델을 만든다는 것은 바로 전체 데이터의 잔차 합이 최소가 되는 모델을 만든다는 의미임. 동시에 오류 값 합이 최소가 될 수 있는 최적의 회귀 계수를 찾는다는 의미도 됨.
=> 절대값(Mean Absolute Error), 제곱(RSS)

RSS는 회귀식의 독립변수 X, 종속변수 Y가 중심 변수가 아니라 w변수가 중심 변수임을 인지하는 것이 매우 중요함.

[5.3] 비용 최소화하기 – 경사하강법 소개

경사하강법 : 점진적으로 반복적인 계산을 통해 W 파라미터 값을 업데이트하면서 오류 값이 최소가 되는 W 파라미터를 구하는 방식.
(단점) : 모든 학습 데이터에대해 반복적으로 비용함수 최소화르 루이한 값을 업데이트하기 때문에 수행 시간이 매우 오래 걸림. -> 확률적 경사 하강법 이용.

확률적 경사 하강법 : 전체 입력 데이터로 w가 업데이트되는 값을 계산하는 것이 아니라 일부 데이터만 이용해 w가 업데이트되는 값을 계산하므로 경사 하강법에 비해서 빠른 속도를 보장함.
차이 : 전체 X, y 데이터에서 랜덤하게 batch_size만큼 데이터를 추출해 이를 기반으로 w1_update, w0_update를 계산하는 부분만 차이가 있음.

대용량의 데이터의 경우 대부분 확률적 경사 하강법이나 미니 배치 확률적 경사 하강법을 이용해 최적 비용 함수를 도출함.

[5.4] 사이킷런 LinearRegression을 이용한 보스턴 주택 가격 예측

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model

cf. (갑자기 까먹어서..) n_jobs : 매개 변수를 사용하여 사용할 코어 수

LinearRegression 클래스 :
예측값과 실제 값의 RSS를 최소화해 OLS 추정 방식으로 구현한 클래스.
fit() 메서드로 X,y 배열을 입력받으면 회귀 계수인 W를 coef_속성에 저장함.

Ordinary Least Squared 기반의 회귀 계산은 입력 피처의 독립성에 많은 영향을 받음. 피처 간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 매우 민감함. 이러한 현상을 다중 공선성 문제라고 함. 일반적으로 상관관계가 높은 피처가 많은 경우 독립적인 중요한 피처만 남기고 제거하거나 규제를 적용함. 또한 매우 많은 피처가 다중 공선성 문제를 가지고 있다면 PCA를 통해 차원 축소를 수행하는 것도 고려해볼 수 있음.

MAE, MSE, RMSE, R^2, (MSLE, RMSLE)

[5.5] 다항 회귀와 과(대)적합/ 과소적합 이해
회귀가 독립변수의 단항식이 아닌 2차, 3차 방정식과 같은 다항식으로 표현되는 것을 다항 회귀라고 함. 다항회귀는 선형 회귀임.
(회귀에서 선형 회귀/ 비선형회귀를 나누는 기준은 회귀 계수가 선형/비선형인지에 따른 것이지 독립 변수의 선형/ 비선형 여부와는 무관함)

다항 회귀는 피처의 직선적 관계가 아닌 복잡한 다항 관계를 모델링할 수 있음.
다항식의 차수가 높아질수록 매우 복집한 피처 간의 관계까지 모델링이 가능함. 하지만, 다항 회귀의 차수를 높일수록 학습 데이터에만 너무 맞춘 학습이 이뤄져서 정작 테스트 데이터 환경에서는 오히려 예측 정확도가 떨어짐. 즉, 차수가 높아질수록 과적합의 문제가 크게 발생함.

저편향/ 저분산 : 예측 결과가 실제 결과에 매우 잘 근접하면서도 예측 변동이 크지 않고 특정 부분에 집중돼 있는 아주 뛰어난 성능을 보여줌.
저편향 / 고분산 : 예측 결과가 실제 결과에 비교적 근접하지만, 예측 결과가 실제 결과를 중

심으로 꽤 넓은 부분에 분포.
고편향/저분산 : 정확한 결과에서 벗어나면서도 예측이 특정 부분에 집중돼 있음.
고편향 / 고분산 : 정확한 예측 결과를 벗어나면서도 넓은 부분에 분포되어 있음.
=> 일반적으로 편향과 분산은 한쪽이 높으면 한쪽이 낮아지는 경향이 있음.
골다익스 : 편향을 낮추고 분산을 높이면서 전체 오류가 가장 낮아지는 지점.

높은 편향 / 낮은 분산에서 과적합되기 쉬우며, 낮은 편향/높은 분산에서 과적합되기 쉬움.
편향과 분산이 서로 트레이드오프를 이루면서 오류 Cost 값이 최대로 낮아지는 모델을 구하는 것이 가장 효율적인 머신러닝 예측 모델을 만드는 방법임.

[5.6] 규제 선형 모델 - 릿지, 라쏘, 엘라스틱넷

회귀 모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 함.
-> 비용 함수는 학습 데이터의 잔차 오류 값을 최소로 하는 RSS 최소화하는 방법과 과적합을 방지하는 위해 회귀 계수 값이 커지지 않도록 하는 방법이 서로 균형을 이뤄야 함.

alpha : alpha 값을 크게 하면 비용 함수는 회귀 계수 W의 값을 작게 해 과적합을 개선할 수 있으며 alpha 값을 작게 하면 회귀 계수 W의 값이 커져도 어느 정도 상쇄가 가능하므로 학습 데이터 적합을 더 개선할 수 있음.

규제 : 비용 함수에 alpha 값으로 페널티를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식. 크게 L2 방식과 L1 방식으로 구분됨.
릿지 회귀 : L2 규제(W의 제곱에 대해 페널티를 부여) 적용 회귀
라쏘 회귀 : L1 규제(W의 절댓값에 대해 패널티를 부여. 영향력이 크지 않은 회귀 계수 값을 0으로 변환)를 선형 회귀에 적용한 회귀.

릿지 회귀:
사이킷런은 Ridge 클래스를 통해 릿지 회귀를 구현함. Ridege 클래스의 주요 생성 파라미터는 alpha이며, 이는 릿지 회귀의 alpha L2 규제 계수에 해당함.
릿지 회귀는 alpha 값이 커질수록 회귀 계수 값을 작게 만듬. 하지만, 릿지 회귀의 경우에는 회귀 계수를 0으로 만들지는 않음.

라쏘 회귀 :
W의 절댓값에 페널티를 부여하는 L1 규제를 선형 회귀에 적용한 것이 라쏘 회귀임.
L2 규제가 회귀 계수의 크기를 감소시키는데 반해, L1 규제는 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만들고 제거함. 이러한 측면에서 L1 규제는 적절한 피처만 회귀에 포함시키는 피처 선택의 특성을 가지고 있음.
사이킷런은 Lasso 클래스를 통해 라쏘 회귀를 구현함. Lasso 클래스의 주요 생성 파라미터는 alph이며, 이는 라쏘 회귀의 alpha L1 규제 계수에 해당함.

엘라스틱넷 회귀 :
L2규제와 L1 규제를 결합한 회귀.
라쏘 회귀가 서로 상관관계가 높은 피처들의 경우에 이들중에서 중요 피처만을 셀렉션하고 다른 피처들은 모두 회귀 계수를 0으로 만드는 성향이 강함. 특히 이러한 성향으로 인해 alpha 값에 따라 회귀 계수의 값이 급격히 변동할 수도 있는데, 엘라스틱넷 회귀는 이를 완화하기 위해 L2 규제를 라쏘 회귀에 추가한 것임.
(단점) : 수행시간이 상대적으로 오래 걸림.
사이킷런은 ElasticNet 클래스 사용. ElasticNet 클래스의 주요 생성 파라미터는 alpha와 l1_ratio임.(ElasticNet 클래스의 alpha는 Ridge와 ㄹㄴ내 클래스의 alpha 값과는 다름)

상황에 따라 어떤 것이 가장 좋은지 다름. 각각의 알고리즘에서 하이퍼 파라미터를 변경해 가면서 최적의 예측 성능을 찾아내야 함. 하지만, 선형 회귀의 경우 최적의 하이퍼 파라미터를 찾아내는 것 못지않게 먼저 데이터 분포도의 정규화와 인코딩 방법이 매우 중요함.

● 선형 회귀 모델을 위한 데이터 변환
선형 회귀 모델과 같은 선형 모델은 일반적으로 피처와 타깃값 간에 선형의 관계가 있다고 가정하고, 이러한 최적의 선형함수를 찾아내 결괏값을 예측함. 또한 선형 회귀 모델은 피처값과 타깃값의 분포가 정규 분포 형태를 매우 선호함.
특히 타깃값의 경우 정규 분포 형태가 아니라 특정값의 분포가 치우친 왜곡된 형태의 분포도일 경우 예측 성능에 부정적인 영향을 미칠 가능성이 높음. 피처값 역시 결정값보다는 덜하지만, 왜곡된 분포도로 인해 예측 성능에 부정적인 영향을 미칠 수 있음.
따라서!! 선형 회귀 모델을 적용하기 전에 먼저 데이터에 대한 스케일링/정규화 작업을 수행하는 것이 일반적임. 하지만, 스케일링/정규화 작업을 선행한다고 해서 무조건 예측 성능이 향상되는 것은 아님. 일반적으로 중요한 피처들이나 타깃값의 분포도가 심하게 왜곡됐을 경우에 이러한 변환 작업을 수행함.
피처 데이터 세트에 스케일링/ 정규화 작업 수행 방법
1. StandardScaler 클래스를 이용해 평균이 0, 분산이 1인 표준 정규 분포를 가진 데이터 세트로 변환하거나 MinMaxScaler 클래스를 이용해 최솟값이 0이고 최댓값이 1인 값으로 정규화를 수행.
2. 스케일링/정규화를 수행한 데이터 세트에 다시 다항 특성을 적용하여 변환하는 방법. 보통 1번 방법을 통해 예측 성능에 향상이 없을 경우 이와 같은 방법을 적용함.
3. 원래 값에 log 함수를 적용하면 보다, 정규 분포에 가까운 형태로 값이 분포됨.(로그 변환) 1,2번 방법보다 많이 사용되는 변환 방법임. 1번 방법의 경우 예측 성능 향상을 크게 기대하기 어려운 경우가 많으며 2번 방법의 경우 피처의 개수가 매우 많을 경우에는 다항 변환으로 생성되는 피처의 개수가 기하급수로 늘어나서 과적합의 이슈가 발생할 수 있음.

로그 변환에서 np.log1p() 사용 이유 : 일반적으로 log() 함수를 적용하면 언더 플로우가 발생하기 쉬워서 1+log() 함수를 적용하는데, 이를 구현한 것이 np.log1p()임.

d5.7 로지스틱 회귀

로지스틱 회귀는 선형 회귀 방식을 분류에 적용한 알고리즘. 즉, 로지스택 회귀는 분류에 사용됨. 로지스틱 회귀 역시 선형 회귀 계열임.
*회귀가 선형인가 비선형인가는 독립변수가 아닌 가중치 변수가 선형인지 아닌지를 따름.
로지스틱 회귀가 선형 회귀와 다른 점은 학습을 통해 선형 함수의 회귀 최적선을 찾는 것이 아니라 시그모이드 함수 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류를 결정한다는 것임.

로지스틱 회귀는 선형 회귀 방식을 기반으로 하되 시그모이드 함수를 이용해 분류를 수행하는 회귀임.

사이킷런은 로지스틱 회귀를 위해서 LogisticRegression 클래스를 제공함. LogisticRegression 클래스의 회귀 계수 최적화는 경사 하강법 외에 다양한 최적화 방안을 선택할 수 있음. lgbfs, liblinear, newton-cg, sag, saga
-> 다양한 solver 값들이 있지만, 이들간의 성능 차이는 미비하며 일반적으로 lbfgs 또는 liblinear를 선택하는 것이 일반적임.

cf. 선형 회귀 계열의 로지스틱 회귀는 데이터의 정규 분포도에 따라 예측 성능 영향을 받을 수 있음!

파라미터 :
solver : 최적화 방법 선택.
max_iter : solver로 지정된 초적화 알고리즘이 최적 수렴할 수 있는 최대 반복 횟수.
penalty : 규제의 유형을 설정하며, l2로 설정 시 L2 규제를, l1으로 설정시 L1 규제를 뜻함.
C : 규제 강도를 조절하는 alpha 값의 역수. 1/alpha. C값이 작을수록 규제 강도가 큼.

L1, L2 규제의 경우 solver 설정에 따라 영향을 받음. liblinear, saga의 경우 L1, L2 규제가 모두 가능하지만, lbfgs, newton-cg, sag의 경우는 L2 규제만 가능함.

로지스틱 회귀는 가볍고 빠르지만,이진 분류 예측 성능도 뛰어남. 이 때문에 로지스틱 회기를 이진 분류의 기본 모델로 사용하는 경우가 많음. 또한 로지스틱 회귀는 희소한 데이터 세트 분류에도 뛰어난 성능을 보여서 텍스트 분류에서도 자주 사용됨.

5.8 회귀 트리
선형 회귀는 회귀 계수의 관계를 모두 선형으로 가정하는 방식임. 일반적으로 선형 회귀는 회귀 계수를 선형으로 결합하는 회귀 함수를 구해, 여기에 독립 변수를 입력해 결괏값을 예측하는 것임.
비선형 회귀 역시 비선형 회귀 함수를 통해 결괏값을 예측함. 다만, 비선형 회귀는 회귀 계수의 결합이 비선형일 뿐임.
트리 기반의 회귀는 회귀 트리를 이용하는 것임. 즉, 회귀를 위한 트리를 생성하고 이를 기반으로 회귀 예측을 하는 것임.
회귀 트리는 리프 노드에 속한 데이터 값의 평균값을 구해 회귀 예측값을 계산함.

앞 4장의 분류에서 소개한 모든 트리 기반의 알고리즘은 분류뿐만 아니라 회귀도 가능함. 트리 생성이 cart 알고리즘에 기반하고 있기 때문임.

회귀 트리 Regressor 클래스는 선형 회귀와 다른 처리 방식이므로 회귀 계수를 제공하는 coef_ 속성이 없음. 대신 feature_importances_를 이용해 피처별 중요도를 알 수 있음.

*선형 회귀는 직선으로 예측 회귀선을 표현하는 데 반해, 회귀 트리의 경우 분할되는 데이터 지점에 따라 브랜치를 만들면서 계단 형태로 회귀선을 만듬.