Python으로 머신러닝 입문

세션장: 구은아, 김은기

Ridge

Ridge 클래스 및 하이퍼파라미터

sklearn.linear_model.Ridge

class sklearn.linear_model.Ridge(*alpha=1.0, *, fit_intercept=True, normalize='deprecated', copy_X=True, max_iter=None, tol=0.001, solver='auto', positive=False, random_state=None*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

파라미터 이름	설명
alpha	L2 규제 계수 alpha값이 클수록 과적합을 더 강하게 방지한다. 디폴트는 1.0

Lasso

Lasso 클래스 및 하이퍼파라미터

sklearn.linear_model.Lasso

class sklearn.linear_model.Lasso(*alpha=1.0, *, fit_intercept=True, normalize='deprecated', precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

파라미터 이름	설명
alpha	L1 규제 계수 alpha값이 클수록 과적합을 더 강하게 방지한다. 디폴트는 1.0

ElasticNet

ElasticNet 클래스 및 하이퍼파라미터

sklearn.linear_model.ElasticNet

class sklearn.linear_model.ElasticNet(*alpha=1.0, *, l1_ratio=0.5, fit_intercept=True, normalize='deprecated', precompute=False, max_iter=1000, copy_X=True, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

파라미터 이름	설명
alpha	L1과 L2의 규제 계수 alpha값이 클수록 과적합을 더 강하게 방지한다. 디폴트는 1.0
l1_ratio	L1 규제치를 반영할 비율(a/(a+b)에서 a) 디폴트는 0.5

로지스틱 회귀

로지스틱 회귀 클래스 및 하이퍼파라미터

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.LogisticRegression(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

파라미터 이름	설명
penalty	사용할 규제의 종류(l1: lasso, l2: ridge) 디폴트는 l2
C	규제 계수 alpha의 역수 디폴트는 1.0

트리기반 회귀

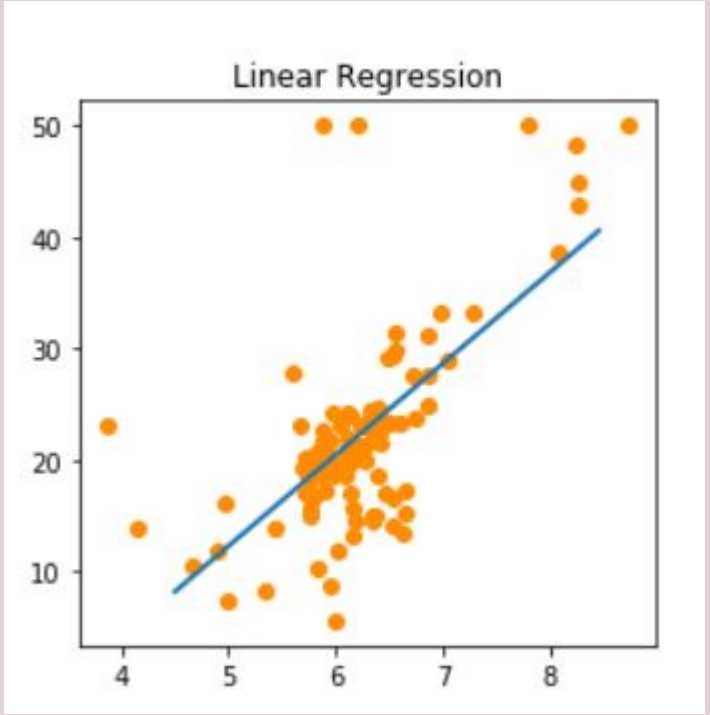
트리기반 회귀 클래스

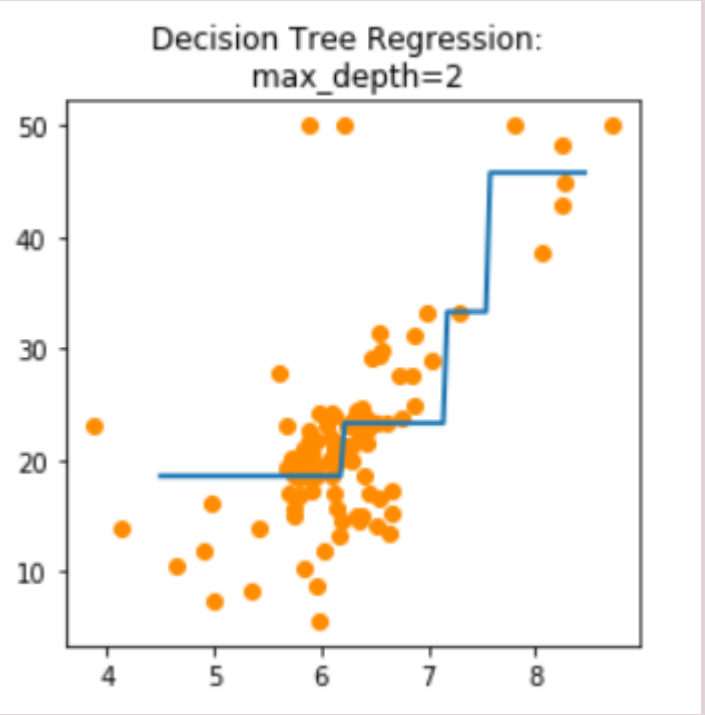
결정트리 기반
DecisionTreeRegressor

랜덤포레스트 기반
RandomForestRegressor

XGBoost 기반
XGBRegressor

LightGBM 기반
LGBMRegressor

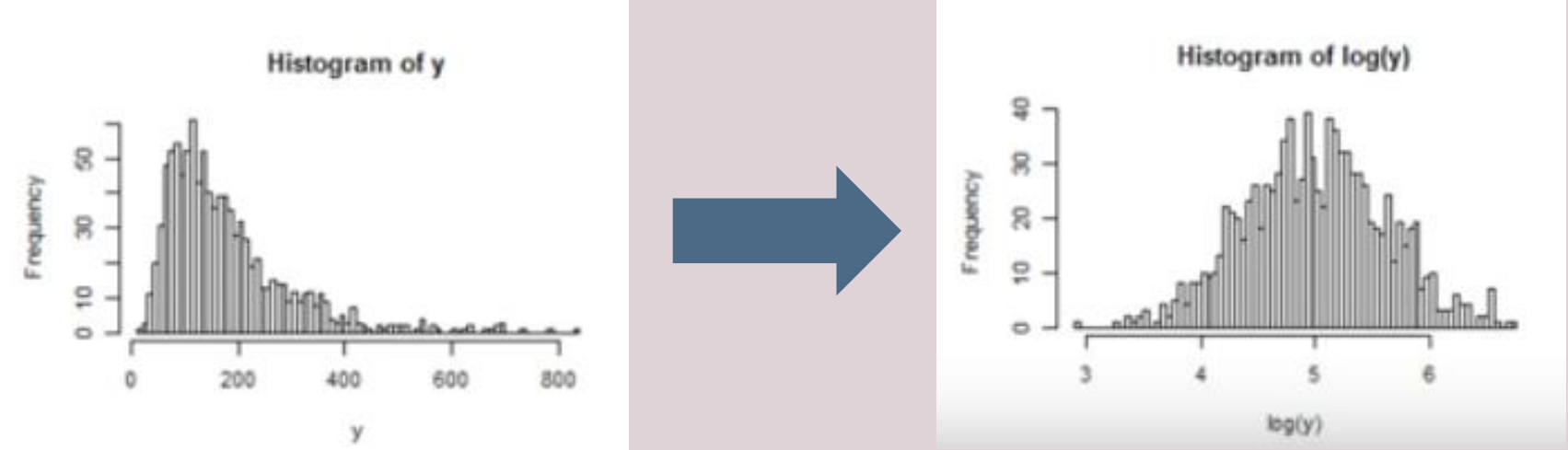




유용한 데이터 변환

스케일링/정규화

함수명	설명
StandardScaler (사이킷런 제공)	평균이 0, 분산이 1인 표준정규분포를 따르도록 데이터를 표준화(정규화) 데이터의 분산을 통일화(스케일링)
np.log1p	데이터에 로그 함수를 적용하여 정규화 복원할 때는 np.expm1 사용



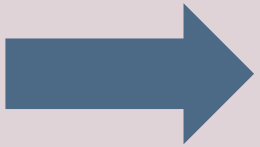
라벨 인코딩

sklearn.preprocessing.LabelEncoder

class sklearn.preprocessing.LabelEncoder [source]

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

단과대
문과대
정경대
정경대
경영대
문과대
경영대



문과대: 1
정경대: 2
경영대: 3

단과대_인코딩
1
2
2
3
1
3

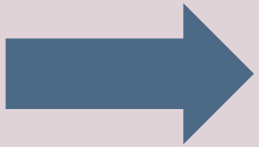
원핫 인코딩

sklearn.preprocessing.OneHotEncoder

class sklearn.preprocessing.OneHotEncoder(*, *categories='auto'*, *drop=None*, *sparse=True*, *dtype=<class 'numpy.float64'>*, *handle_unknown='error'*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

단과대
문과대
정경대
정경대
경영대
문과대
경영대



문과대	정경대	경영대
1	0	0
0	1	0
0	1	0
0	0	1
1	0	0
0	0	1

과제 안내

과제: 자전거 대여 수요 예측

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32

1) datetime 열에서 연도와 월만 추출하여 다른 열로 만들어봅시다.
2) season, weather는 이미 라벨 인코딩되어 있습니다. 라벨 인코딩과 원핫 인코딩 모두 학습해보고 어떤 것이 더 좋은 성능을 보이는지 판단해봅시다. 그리고 어떤 열을 인코딩시키면 좋을지 생각해보고 적용해봅시다.
3) count 열의 분포는 정규성을 띄지 않습니다. log 변환을 통해 정규화하고, 성능을 평가할 때는 원래 값으로 되돌려봅시다.
4) 오늘 공부했던 모든 회귀 모델을 사용하여 성능을 비교해봅시다. 어떤 모델의 성능이 가장 좋나요? 또한 릿지와 라쏘의 경우, 어떤 alpha 값을 가질 때 성능이 가장 좋나요?

수고하셨습니다!
과제 열심히 하시고 다음 주에 뵈어요~