202263042 홍우진 머신러닝 보고서

# 데이터 불러오기

텍스트, 모니터, 은색, 스크린샷이(가) 표시된 사진

자동 생성된 설명

먼저 데이터를 불러온다. 모델의 성능은 통상적으로 데이터가 많을수록 높아지기 때문에 데이터 두개를 합쳐서 사용하였다.

테이블이(가) 표시된 사진

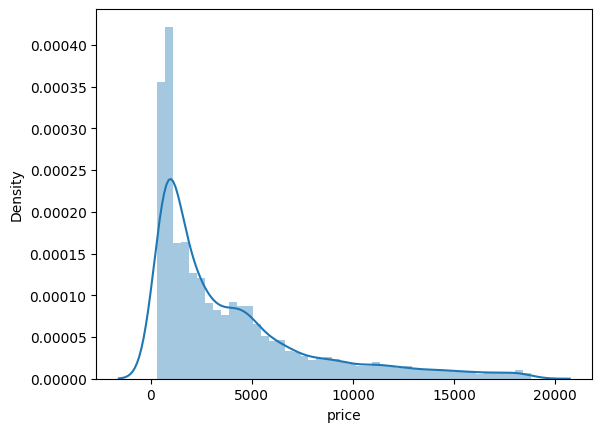
자동 생성된 설명

회귀 모델에 치명적일 수 있는 데이터의 결측치를 확인한다.

# EDA 및 이상치 처리

­­텍스트이(가) 표시된 사진

자동 생성된 설명



먼저 타겟 데이터인 가격의 그래프를 그려본다. 가격대가 높을수록 수가 적어지는 것을 확인 할 수 있다.

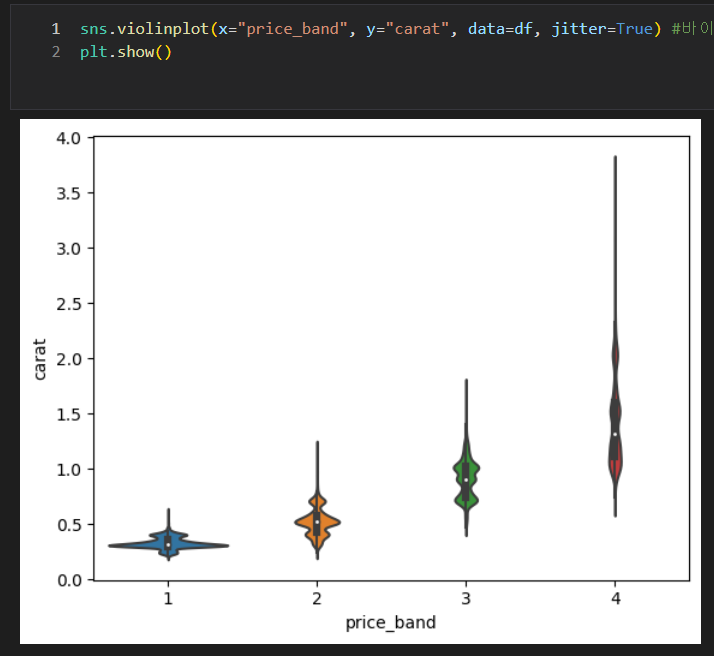
텍스트이(가) 표시된 사진

자동 생성된 설명

가격의 분포도를 확인한다.

텍스트, 모니터, 스크린샷, 검은색이(가) 표시된 사진

자동 생성된 설명

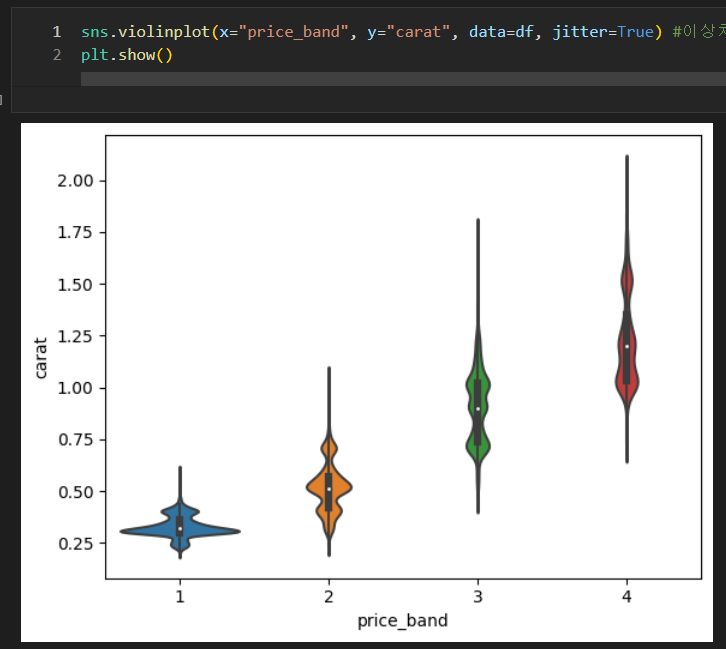
확인한 분포도를 가반으로 가격을 5단계로 나눈 후, 가격이 균일하게 분포되었는지 확인한다.

바이올린플롯으로 분포도를 확인한다. 가격의 분포가 이상함을 확인할 수 있다.

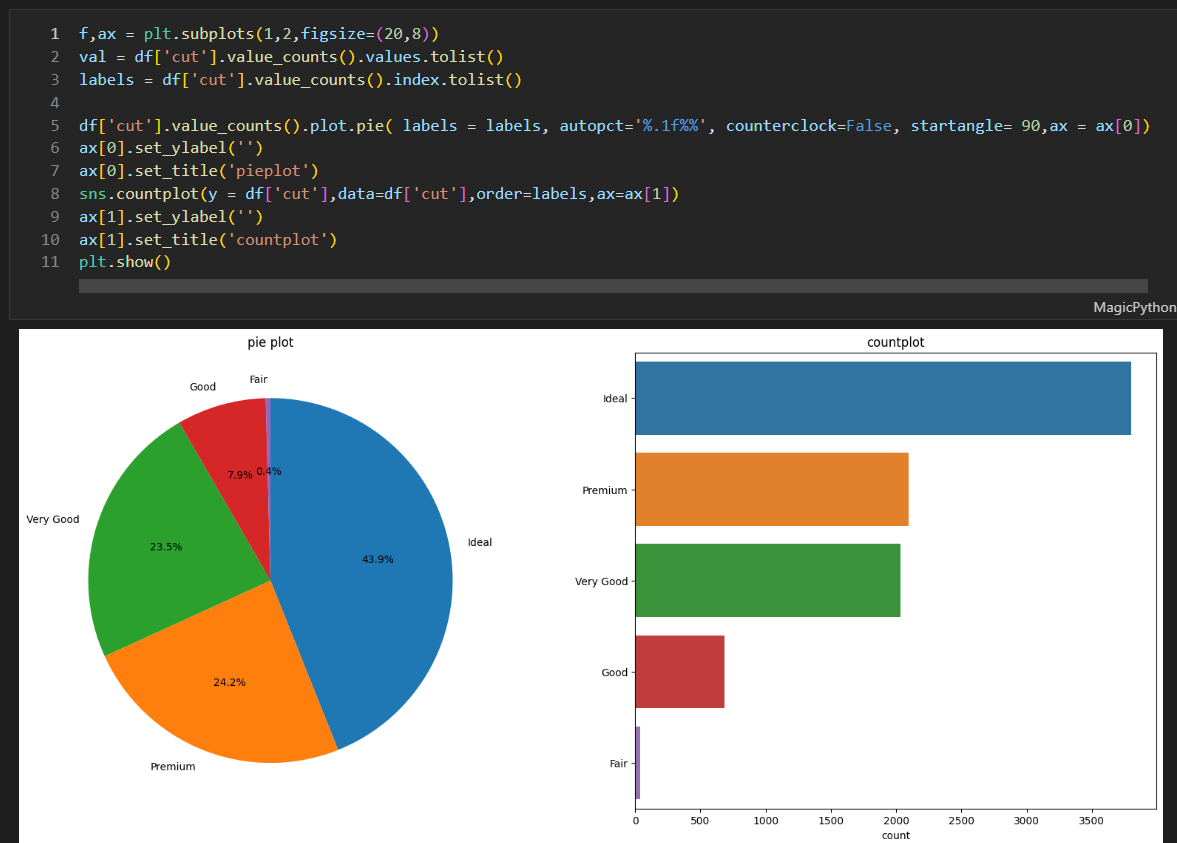
텍스트이(가) 표시된 사진

자동 생성된 설명

IQR방식을 사용하여 연속형 컬럼들의 이상치를 제거하여 준다.



이상치가 처리된 것을 확인할 수 있다.



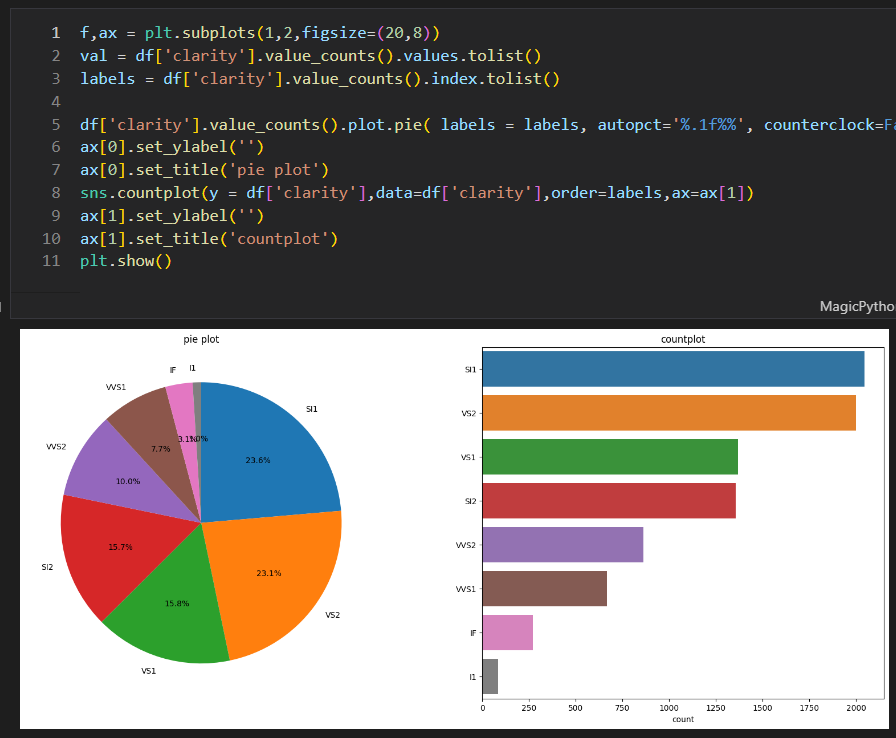
CUT 컬럼 안의 데이터를 pieplot과 countplot으로 확인한다.

'Ideal','Premium','Very Good','Good','Fair' 순으로 많은 것을 확인할 수 있다.

텍스트이(가) 표시된 사진

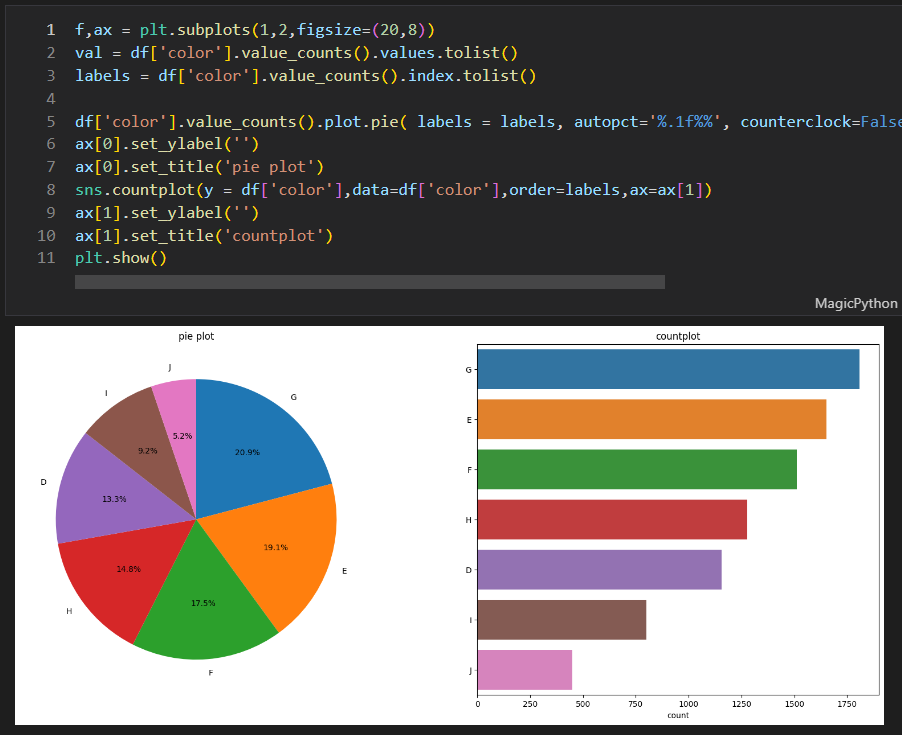
자동 생성된 설명

모델학습을 위하여 비싼 등급순으로 문자를 숫자로 변경하여 준다.



텍스트이(가) 표시된 사진

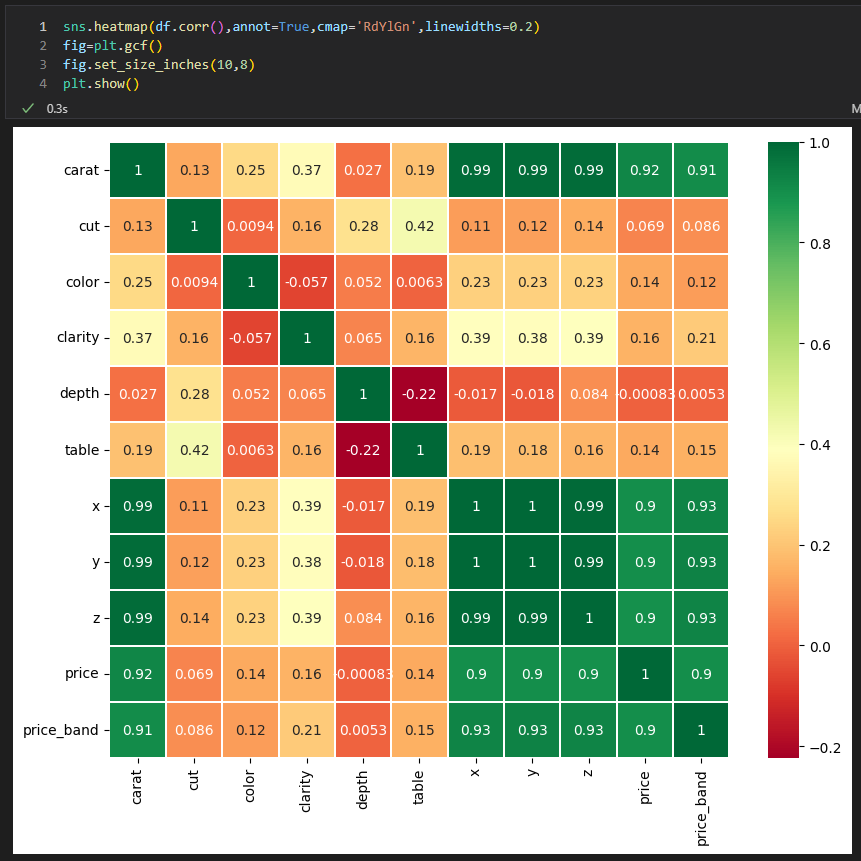
자동 생성된 설명

텍스트이(가) 표시된 사진

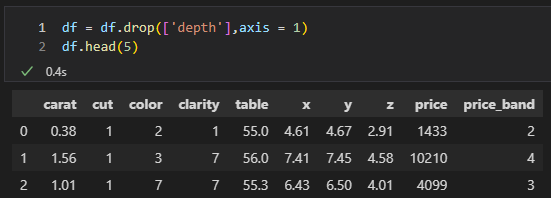
자동 생성된 설명

투명도와 색깔 컬럼도 CUT 컬럼과 같은 과정으로 진행하여 준다.

데이터의 비율을 확인하여 주고, 가격이 높은 등급 순으로 등급으로 변환하여 준다.



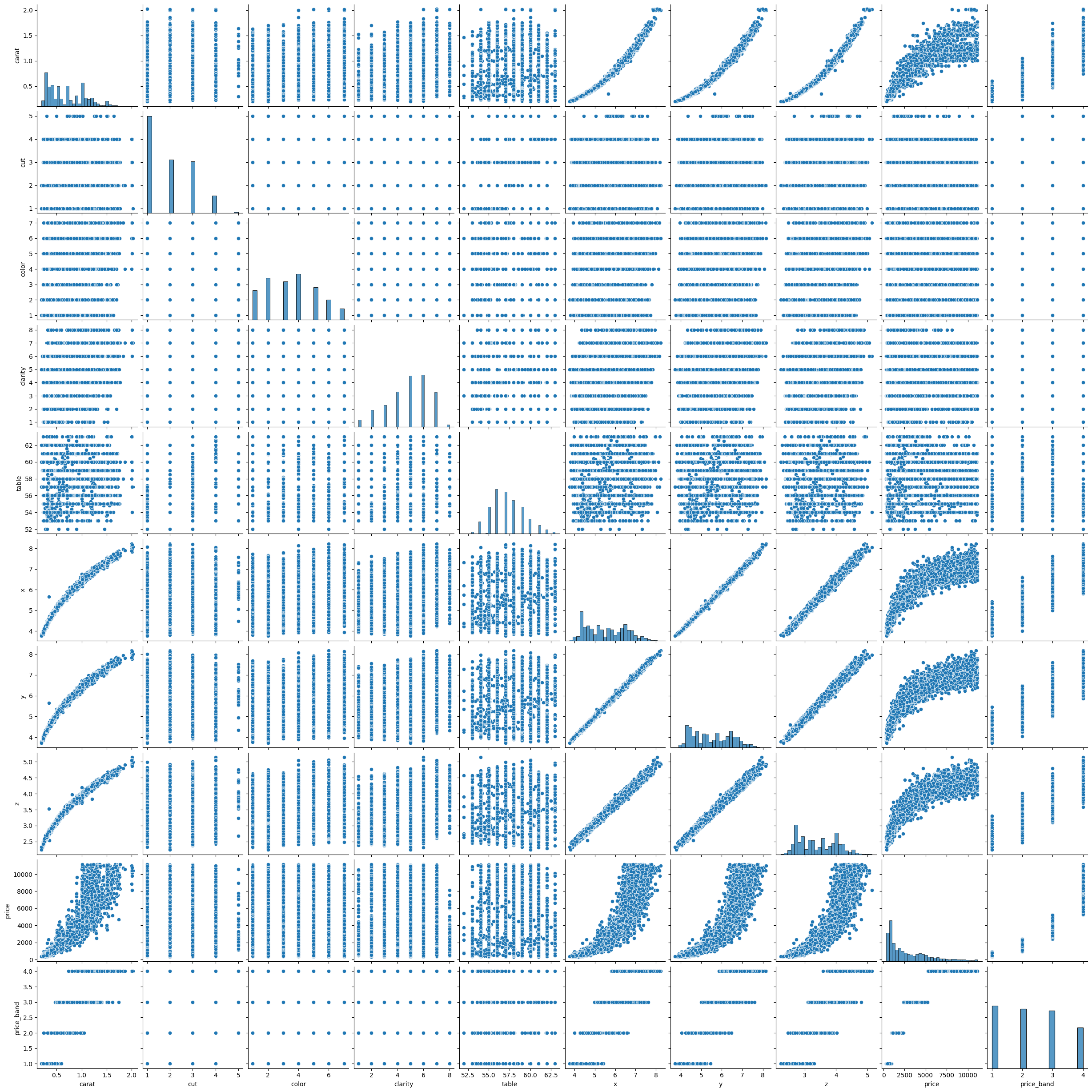
컬럼 간의 상관도 분석을 위하여 Heatmap을 그려준다.



Depth 컬럼이 우리가 예측할 price 컬럼과의 상관도가 매우 낮기 때문에 성능 향상을 위하여 제거한다.

­­텍스트이(가) 표시된 사진

자동 생성된 설명



pairplot으로 데이터 간의 연관성을 확인한다.

# 모델 선정 및 학습

텍스트이(가) 표시된 사진

자동 생성된 설명

Price를 target 값으로 선정하고 train, test 로 split 한다.

다음은 선정한 모델들이다.

1. LinearRegression
2. RidgeRegression
3. KNeighborsRegressor
4. SGDRegressor
5. RandomForestRegressor
6. LGBMRegressor

공부하는 겸 여러 모델을 시도해보았는데 이렇게 많아질 줄은 몰랐다.

텍스트이(가) 표시된 사진

자동 생성된 설명먼저 가장 기본적인 LinearRegression부터 시작해보자.

Test 점수가 92.0점이 나왔다.

다음은 RidgeRegression이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

PolynomialFeatures를 사용하여 최고 차수를 5제곱까지 가진 컬럼을 만들어서 늘린다.

train 점수는 높아졌지만 test점수는 망가진 과대적합 형태를 확인할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

StandardScaler를 사용하여 표준점수로 정규화를 한 뒤 릿지 규제를 사용하고 학습시킨다.

과대적합이 풀린 모습을 확인할 수 있다.

하지만 높은 점수를 이끌어내기 위하여 릿지의 하이퍼 파라미터인 alpha 값을 최적의 alpha 값으로 변경할것이다.

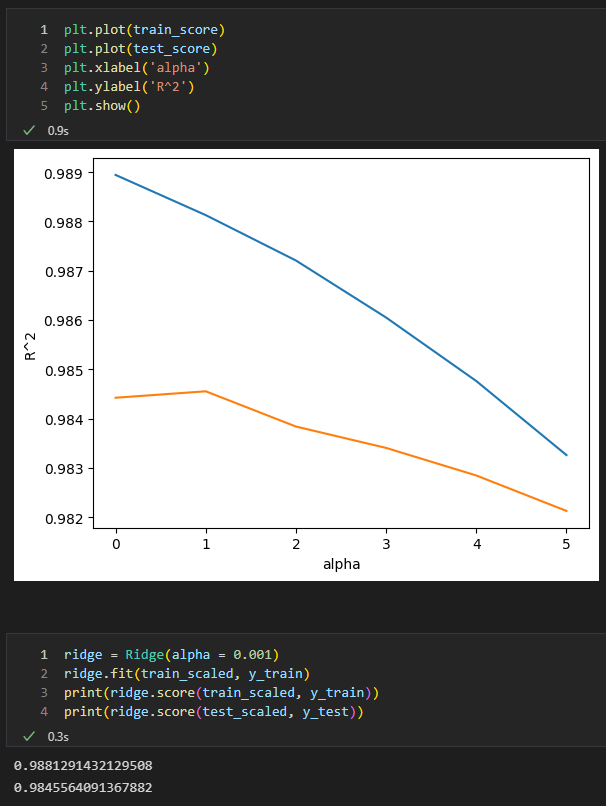
Alpha 값을 변경하면 과적합을 방지하는 규제의 세기를 변경 할 수 있다.

값이 클수록 규제가 세게 들어간다.텍스트, 모니터, 스크린샷, 은색이(가) 표시된 사진

자동 생성된 설명

먼저 점수를 저장할 리스트를 만들어준다.

그리고 alpha 값들이 저장된 리스트를 만들어주고 리스트 안의 값을 이용하여 하나씩 학습시킨 후 점수를 저장하여 준다.

­ 

저장한 점수를 그래프로 그린 후 최적의 alpha값을 선정하고 학습시킨다.

정확도는 98.4점이 나왔다.

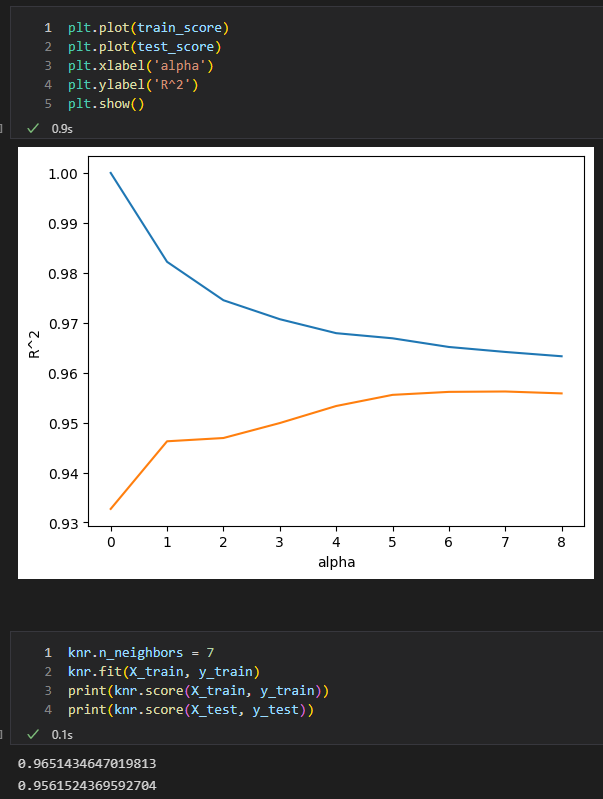
라쏘도 사용을 해보았지만 효과가 좋지 못해 넘어가겠다.

다음은 KNeighborsRegressor이다.

KNeighborsRegressor는 맞추고 싶은 값의 주변 이웃들의 평균을 내서 값을 선정한다.

텍스트, 스크린샷, 은색이(가) 표시된 사진

자동 생성된 설명

Knr도 릿지와 같은 방식을 사용할것인데, 하이퍼 파라미터는 참고할 이웃의 개수로 한다

7개로 선정하였고 점수는 95.6점이 나왔다.

다음은 SGDRegressor 모델을 사용한다.

SGDRegressor는 경사하강법을 사용한다.

경사하강법은 비용함수를 최소화 하기위해 경사를 반복적으로 하강해가면서 파라미터를 조정해 나가는 것이다.

확률적 경사하강법(SGD)은 매 스텝에서 딱 1개의 샘플을 무작위로 선택하고 그에 대한 gradient를 계산한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

SGDRegressor를 사용하여 학습시킨다.

텍스트이(가) 표시된 사진

자동 생성된 설명

점수는 92점이 나왔다.

다음은 RandomForestRegressor이다.

Random Forest는 수많은 Decision Tree가 모여서 생성되는것이다.

Decision Tree는, 특정 Feature 에 대한 질문을 기반으로 데이터를 분리하는 방법이다.

사람들이 일상생활에서 어떠한 의사 결정을 내리는 과정과 매우 비슷하다는 특징이 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

RandomForestRegressor를 학습시킨 후 시각화 한다.

엄청나게 많은 node가 생겼음을 확인 할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

점수는 98.3점이 나왔다.

마지막으로 LGBMRegressor이다.

Light GBM은 트리 기반의 학습 알고리즘인 gradient boosting 방식의 프레임 워크이다.

다른 부스팅 알고리즘에 비해 빠른 학습 속도가 장점이라 선택하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Lgbm은 하이퍼 파라미터가 많기 때문에 GridSearchCV를 사용하였다.

GridSearchCV는 하이퍼 파라미터를 자동으로 grid형식으로 학습시켜서 최적의 하이퍼 파라미터 값을 찾아주는 좋은 라이브러리다.

내가 선택한 하이퍼파라미터는 학습속도, 학습횟수, 행 샘플링이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

최적의 파라미터 값은 'learning\_rate': 0.01, 'n\_estimators': 10000, 'subsample': 0.1 이였고,

점수는 98.3점이 나왔다.

# 최종 모델 선정

최종 선정한 모델은 초반에 학습하였던 Ridge Regression 모델이다.

Test set 정확도 점수가 98.4점으로 다른 모델과 소수점 차이지만 그래도 가장 높았다.

앙상블 모델이나 Forest, Boosting 모델 같은 어려운 모델이 아니라 간단한 규제 모델이 가장 점수가 높게 나왔다는 것이 의외였다.