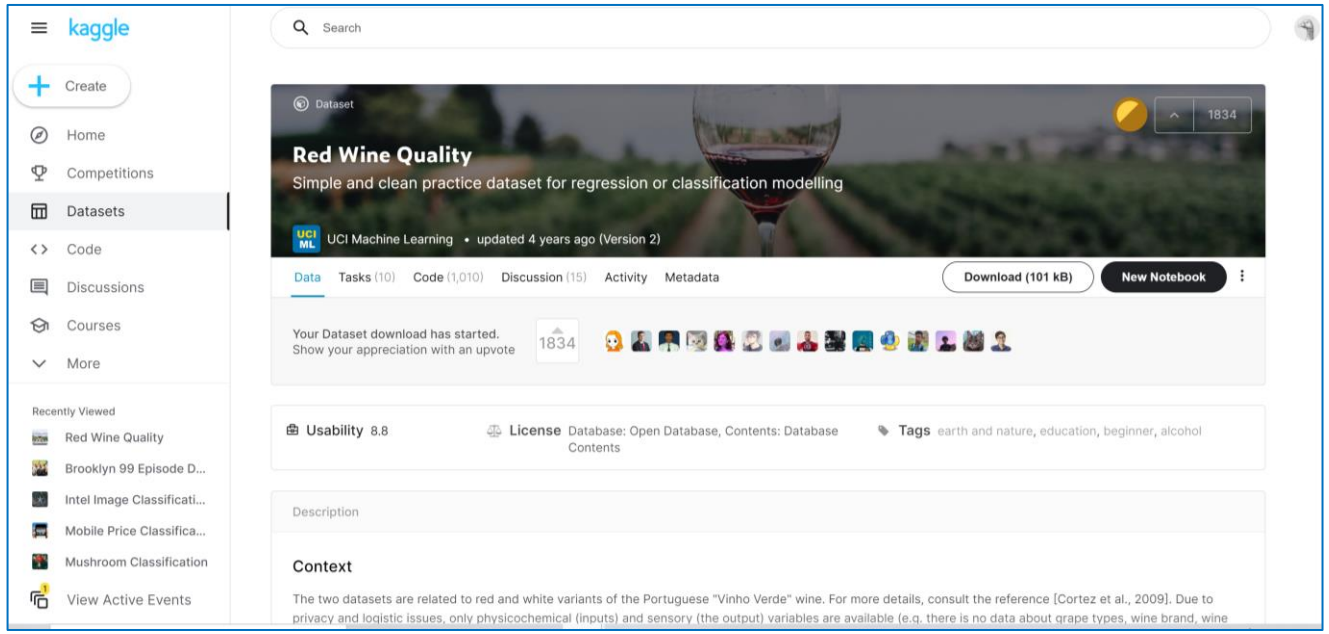
와인 품질 분석

컴퓨터공학전공 19학번 김미경

목차

1. DATASET



총 12개의 변수가 사용됩니다.

fixed acidity, volatile acidity, citric acid,

residual sugar, chlorides,

free sulfur dioxide, total sulfur dioxide

density, ph, sulphates, alcohol,

quality (score between 0 and 10)

quality > 6.5 => good

quality <= 6.5 => bad

기계 학습을 사용해 와인을 'good'으로 만드는 물리화학적 특성을 결정.

1. DATASET

```
wine = pd.read_csv('/kaggle/input/red-wine-quality-cortez-et-al-2009/winequality-red.csv')
wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

데이터 불러오기

총 1599개의 데이터

```
wine.shape
```

```
(1599, 12)
```

```
wine.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	0.135710
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.342587
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	0.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	0.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	0.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	0.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	1.000000

데이터의 전체 통계량을 요약

누락 데이터(NaN)는 제외

1. DATASET

변수	설명	변수	설명
fixed acidity (고정산)	주로 타르타르산(tartaric), 사과산(malic)으로 이루어져 있고 와인의 산도를 제어한다.	**황 화합물 (free sulfur dioxide, total sulfur dioxide, sulphates)**	황 화합물은 원하지 않는 박테리아와 효모를 죽여서 와인을 오래 보관하는 역할
volatile acidity (휘발산)	와인의 향에 연관이 많다.	**density (밀도)**	바디의 높고 낮음을 표현하는 와인의 무게감을 의미한다.
citric acid (구연산)	와인의 신선함을 올려주는 역할, 산성화에 연관을 미친다.	**ph**	와인의 신맛의 정도를 나타낸다.
residual sugar (잔여 설탕)	와인의 단맛을 올려준다.	**alcohol**	와인의 단맛을 주며 와인의 바디감에 영향을 준다.
chlorides (염화물)	와인의 짠맛의 원인이며 와인의 신맛을 좌우하는 성분	**quality**	출력 변수 (0에서 10 사이의 점수)

```
dub_wine=wine.copy()
dub_wine.drop_duplicates(subset=None,inplace=True)
```

```
dub_wine.shape
```

```
wine=dub_wine
```

```
wine.shape
```

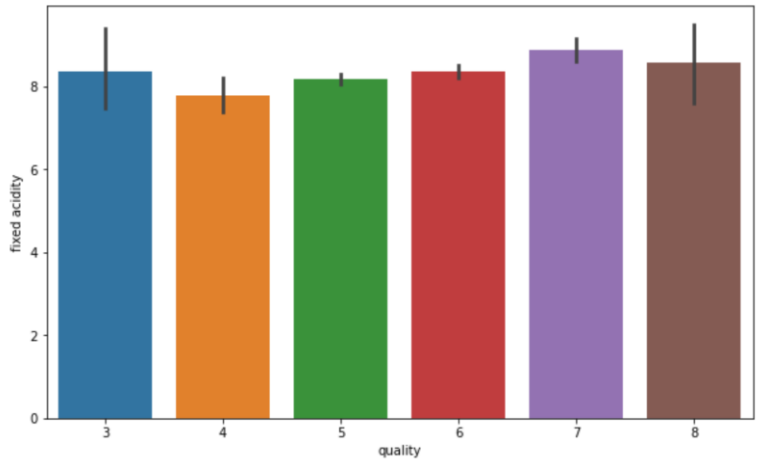
```
(1359, 12)
```

누락 데이터(NaN) 제외
예측의 정확성을 높여줌

2. 데이터 분석 및 시각화

```
#Here we see that fixed acidity does not give any specification to classify the quality.
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'fixed acidity', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='fixed acidity'>
```

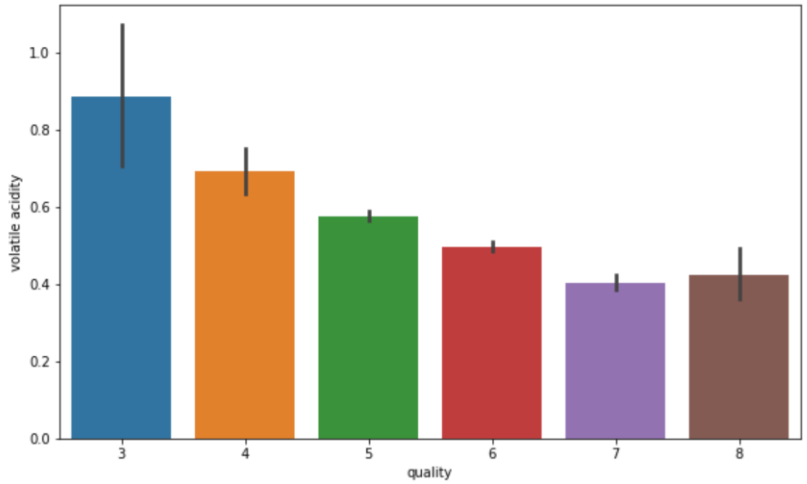


퀄리티와 고정산 연관성

퀄리티와 특정한 상관관계를 가지지 않는다.

```
#Here we see that its quite a downing trend in the volatile acidity as we go higher the quality
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'volatile acidity', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='volatile acidity'>
```



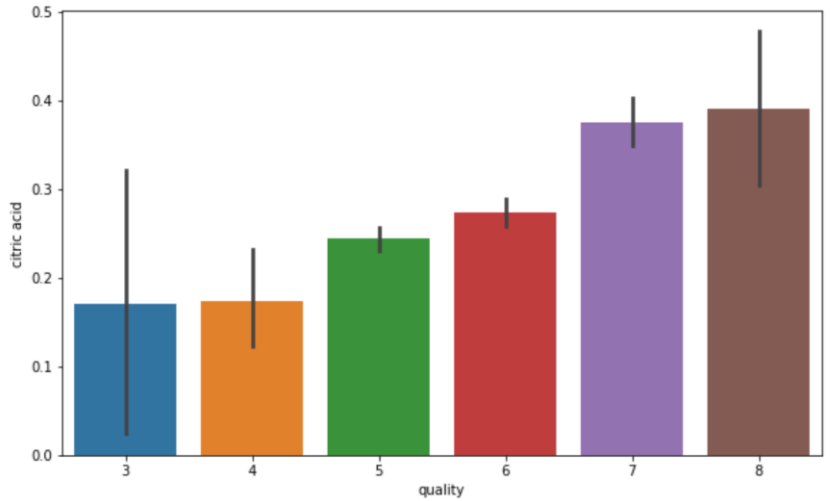
퀄리티와 휘발성 산도의 연관성

품질이 높을 수록 휘발성 산도가 상당히 낮아짐을 알 수 있다.

2. 데이터 분석 및 시각화

```
#Composition of citric acid go higher as we go higher in the quality of the wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'citric acid', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='citric acid'>
```

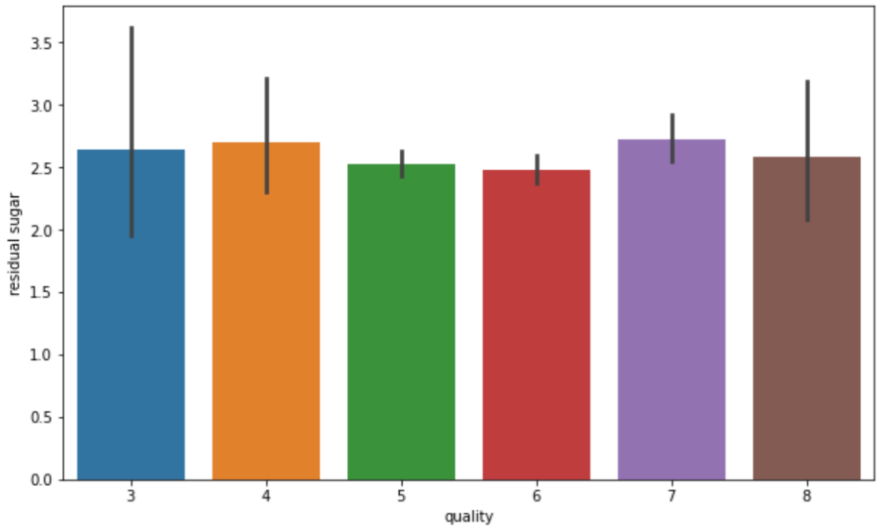


품질과 구연산의 연관성

와인의 품질이 높을수록 구연산의 조성이 높아진다.

```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='residual sugar'>
```



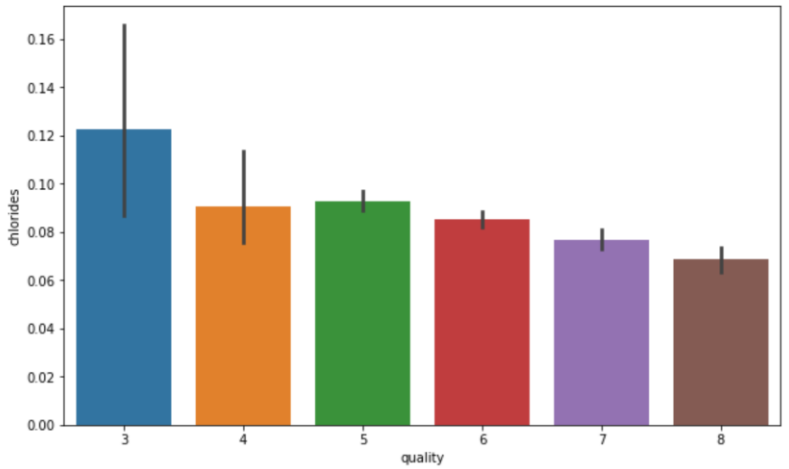
품질과 잔여 당의 연관성

특별한 연관성을 가지지 않는다.

2. 데이터 분석 및 시각화

```
#Composition of chloride also go down as we go higher in the quality of the wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'chlorides', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='chlorides'>
```

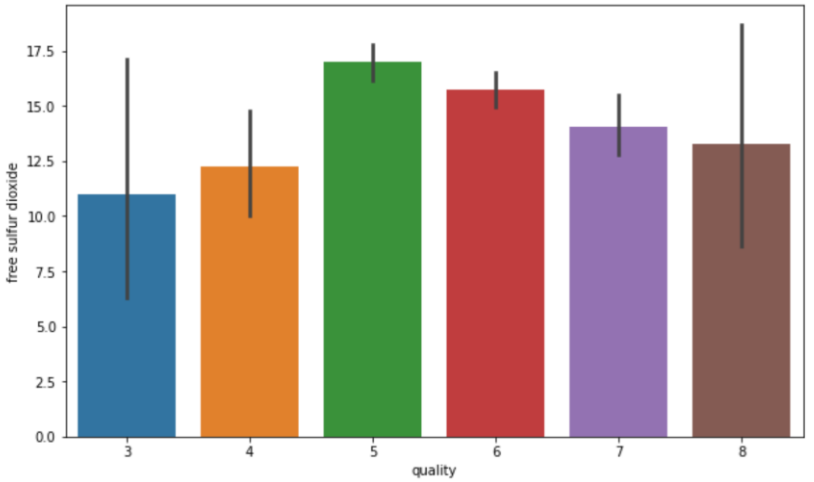


품질과 염화물의 연관성

와인의 품질이 높을수록 염화물의 조성도 낮아진다.

```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='free sulfur dioxide'>
```



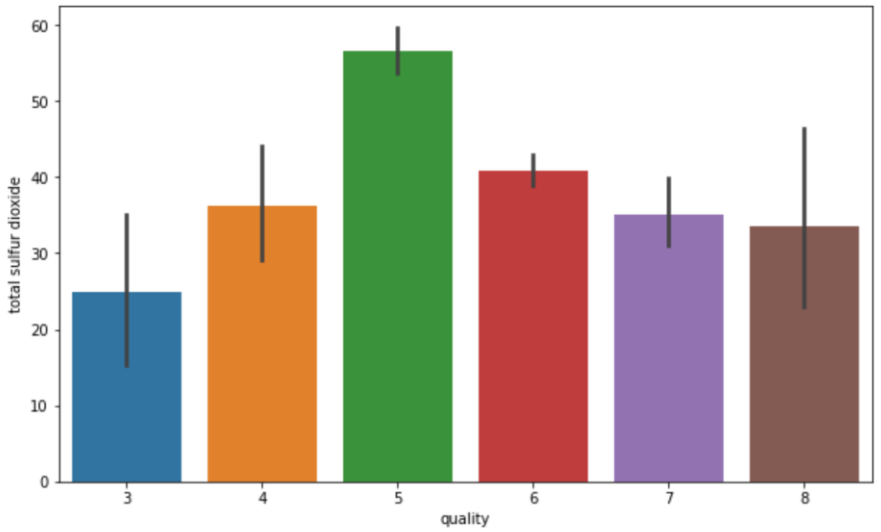
품질과 자유 이산화황의 연관성

특별한 연관성을 가지지 않는다.

2. 데이터 분석 및 시각화

```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'total sulfur dioxide', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='total sulfur dioxide'>
```

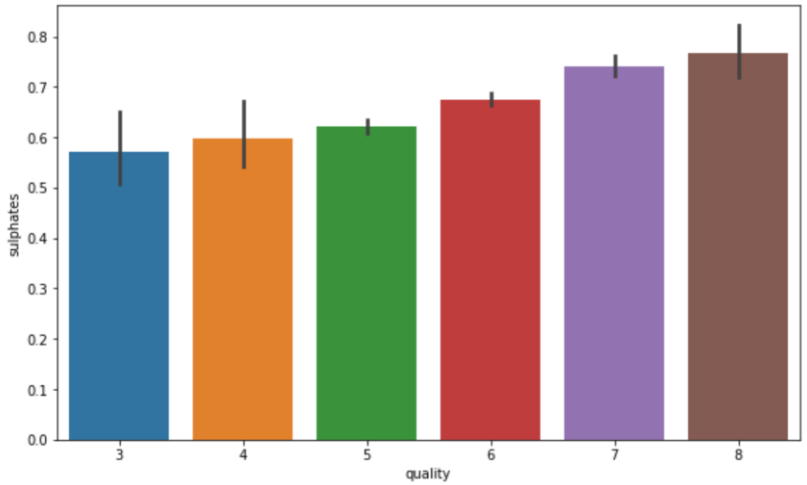


품질과 총 이산화황의 연관성

특별한 연관성을 지니지 않는다.

```
#Sulphates level goes higher with the quality of wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'sulphates', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='sulphates'>
```



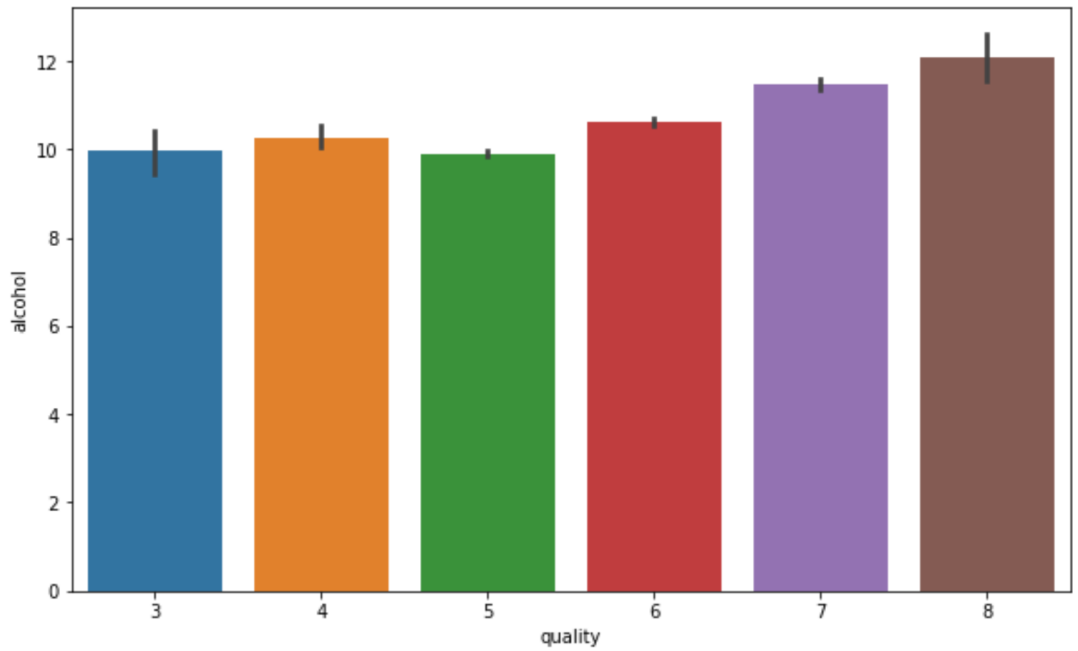
품질과 항산화제의 연관성

와인의 품질이 높을수록 항균 정도가 높아진다.

2. 데이터 분석 및 시각화

```
#Alcohol level also goes higher as te quality of wine increases
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'alcohol', data = wine)
```

```
<AxesSubplot:xlabel='quality', ylabel='alcohol'>
```

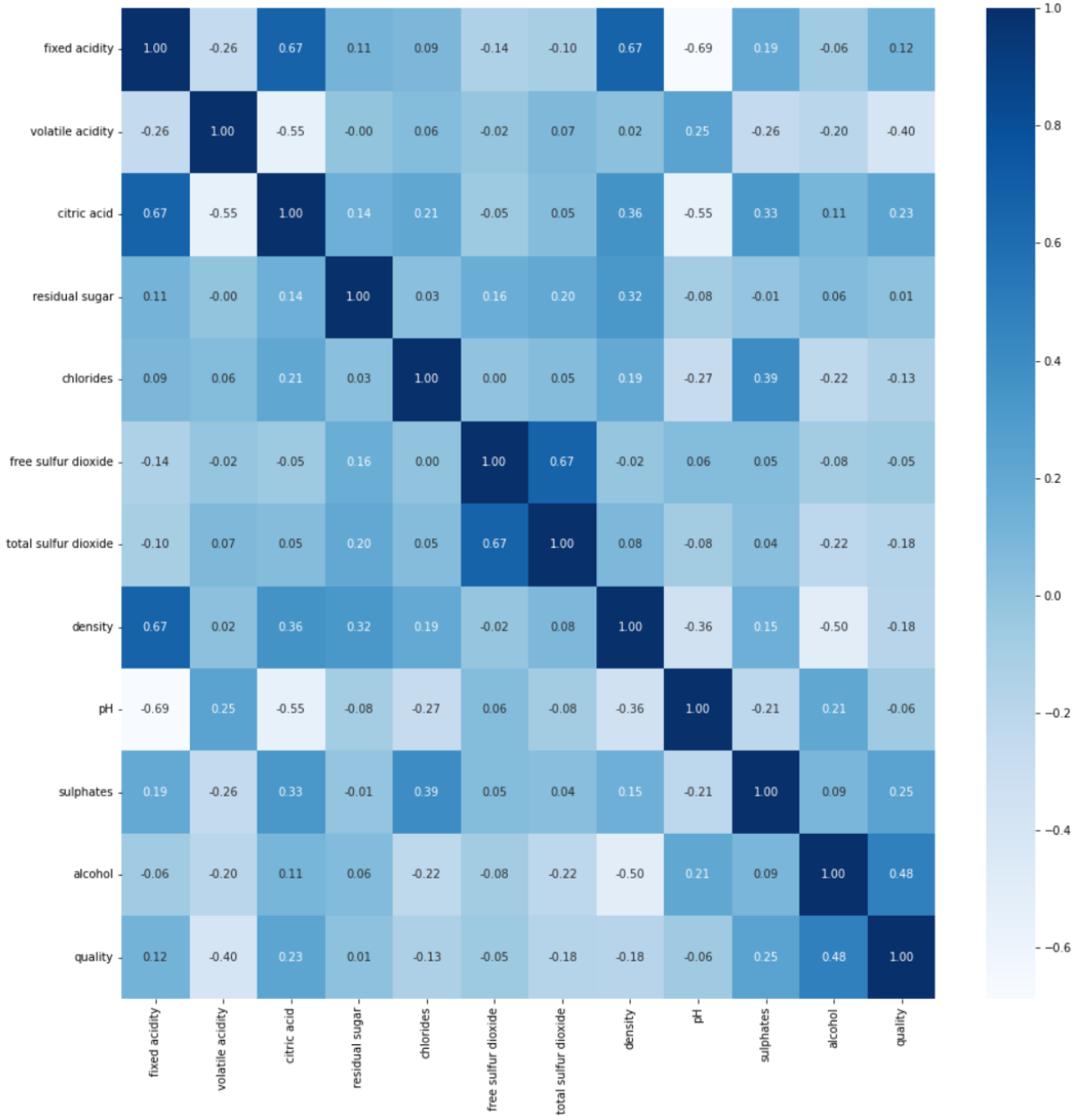


품질과 알코올 농도의 연관성

와인의 품질이 높을수록 알코올 농도도 높아진다.

2. 데이터 분석 및 시각화

```
plt.figure(figsize=(16,16))
sns.heatmap(wine.corr(), cmap='Blues', annot=True, fmt='.2f');
```



Quality와 높은 연관성을 갖는 변수

1. Alcohol (알코올)
2. Volatile acidity (휘발산)

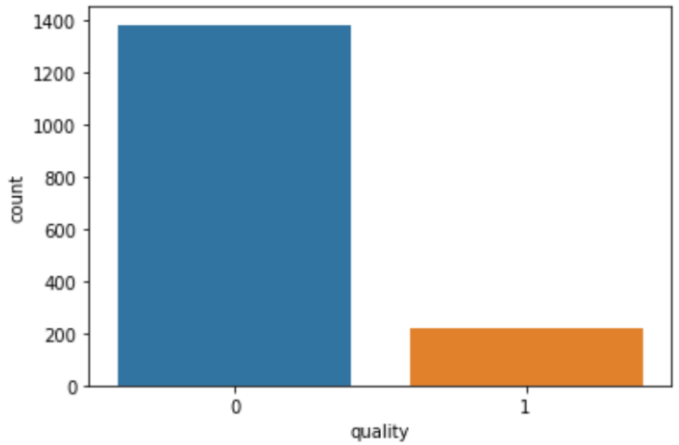
3. EDA

```
#Now lets assign a labels to our quality variable
label_quality = LabelEncoder()

#Making binary classificaion for the response variable.
#Dividing wine as good and bad by giving the limit for the quality
bins = (2, 6.5, 8)
group_names = ['bad', 'good']
wine['quality'] = pd.cut(wine['quality'], bins = bins, labels = group_names)

#Bad becomes 0 and good becomes 1
wine['quality'] = label_quality.fit_transform(wine['quality'])
```

```
sns.countplot(wine['quality'])
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
e `data`, and passing other arguments without an explicit keyword will result in
  FutureWarning
```

```
<AxesSubplot:xlabel='quality', ylabel='count'>
```



quality > 6.5 => good (좋은 품질)

Quality <= 6.5 => bad (보통 품질)

좋은 품질의 데이터보다 보통 품질의 데이터가 많음

```
#Now seperate the dataset as response variable and feature variabes
X = wine.drop('quality', axis = 1)
y = wine['quality']

#Train and Test splitting of data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

#Applying Standard scaling to get optimized result
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

```
X_train.shape
```

```
(1087, 11)
```

```
X_test.shape
```

```
(272, 11)
```

학습데이터와 예측 데이터를 분류

4. 예측

Support Vector Classifier

```
svc = SVC()
svc.fit(X_train, y_train)
pred_svc = svc.predict(X_test)
rate1 = metrics.accuracy_score(pred_svc, y_test)
print('인식률: {0:.4f}'.format(rate1))
```

인식률: 0.9191

92%의 정확도를 가짐

```
print(classification_report(y_test, pred_svc))
```

```
              precision    recall  f1-score   support

           0       0.93      0.99      0.96       238
           1       0.83      0.44      0.58        34

    accuracy                           0.92       272
   macro avg       0.88      0.71      0.77       272
weighted avg       0.91      0.92      0.91       272
```

Support vector classifier gets 92%

4. 예측

Random Forest Classifier

```
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
pred_rfc = rfc.predict(X_test)
rate1 = metrics.accuracy_score(pred_rfc, y_test)
print('인식률: {0:.4f}'.format(rate1))
```

인식률: 0.9191

92%의 정확도를 가짐

```
#Let's see how our model performed
print(classification_report(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.93      0.99      0.96       238
           1       0.83      0.44      0.58        34

    accuracy                           0.92       272
   macro avg       0.88      0.71      0.77       272
weighted avg       0.91      0.92      0.91       272
```

Random forest gives the accuracy of 92%

4. 예측

SGD Classifier

```
sgd = SGDClassifier(penalty=None)
sgd.fit(X_train, y_train)
pred_sgd = sgd.predict(X_test)
rate2 = metrics.accuracy_score(pred_sgd, y_test)
print('인식률: {0:.4f}'.format(rate2))
```

```
인식률: 0.8713
```

87%의 정확도를 가짐

```
print(classification_report(y_test, pred_sgd))
```

```
              precision    recall  f1-score   support

           0       0.88      0.99      0.93       238
           1       0.33      0.03      0.05        34

    accuracy                           0.87       272
   macro avg       0.61      0.51      0.49       272
weighted avg       0.81      0.87      0.82       272
```

87% accuracy using stochastic gradient descent classifier

5. 느낀점

데이터를 직접 분석해 봄으로써 이해하지 못했던 부분에 대한 구분과 그에 대한 이해도를 높일 수 있었습니다. 다만, 데이터에서 와인의 품질이 대체로 높지 못해 변수 간의 상관관계가 높지 않았던 점과 유의미한 결과를 얻지 못한 점이 아쉬웠습니다.

수업시간에 배운 분류 머신러닝 이외에도 딥러닝 학습 방법 중 하나인 SGD를 사용해 데이터를 분석해 봄으로써 머신러닝에 대한 흥미가 증가했습니다. 더 다양한 방식을 활용한 머신러닝을 진행해 보고 싶습니다.

참고

- https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine

QnA