AI+X-Deep Learning Final Project

Title: 스페인 와인 품질 예측

Blog: https://github.com/x1axx/AIX-DL-Projects-F2023

Members:

하걸선, 기계공학부, x1axx@hanyang.ac.kr

장우위, 기계공학부, zyw125@hanyang.ac.kr

채야기, 기계공학부, enoctober@hanyang.ac.kr

송패이, 기계공학부, 1162250193@qq.com

I. Proposal

연구 목적: 이 프로젝트의 연구 목적은 파이썬에서 다양한 회귀 분석 모델을 사용하여 스페인 와인의 점수와 가격에 대한 데이터를 목표로 훈련하고 예측하는 것입니다.연구의 주제는 7,500병의 레드와인에 대한 다양한 기본 정보와 채점을 포함하는 Kaggle 웹사이트에서 발견된 스페인 레드와인 품질 데이터 세트입니다.스페인 와인 데이터 세트를 선택한 이유는 한편으로는 팀원들의 관심 때문이기도 하며, 와인은 항상 사람들의 식탁에서 맛있는 술이었고 스페인은 풍부한 와인 문화와 오랜 역사를 가진 유명한 와인 생산지 중 하나였습니다.스페인은 지중해와 대서양의 중간에 있는 이베리아 반도에 위치하고 있으며, 스페인 와인 양조의 역사는 기원전 1100년으로 거슬러 올라가는 페니키아인들이 와인 양조 기술을 도입하여 지금까지 세계에서 포도 재배 면적이 가장 큰 국가 중 하나이지만 와인 생산량은 프랑스와 이탈리아에 이어 세계 3위에 불과합니다.스페인 와인 산업은 전 세계적으로 중요한 위치를 차지하고 있지만 종종 간과됩니다.국내 와인 상인들이 마케팅 전략에 주의를 기울이지 않는 것 외에도 스페인 사람들은 마시는 와인의 품질에 대한 요구 사항이 높지 않아 제품이 주로 저렴한 와인이며 고품질의 고품질 및 고가 제품이 부족합니다.스페인의 와인 산업은 완전한 법규와 생산 지역, 등급 및 유형 보호 시스템을 갖추고 있으며 스페인에서 가장 유명한 와인은 헤레스 지역에서 생산지 보호와 독특한 양조 방법을 가진 셰리주와 바르셀로나 지역의 와이너리에서 주로 2차 발효를 거쳐 가공된 프랑스 샴페인과 유사한 스파클링 와인입니다.따라서 레드와인의 다양한 지표와 점수 및 가격 간의 관계를 예측하는 모델의 수립은 시장 분석 및 소비자 선호도 변화에 매우 중요합니다.

방법 및 예상 결과: 데이터 세트의 구조를 관찰함으로써 제조업체, 와인 이름, 연도, 점수, 평가 수량, 국가, 지역, 유형, 와인 바디 및 산도의 11가지 속성을 포함할 수 있습니다.국가가 모두 스페인이기 때문에 이 속성은 훈련 과정에서 특정 역할을 하지 않으므로 삭제할 수 있습니다.우리는 점수 또는 가격의 두 가지 속성을 훈련의 목표 변수로 선택할 수 있으며, 얻은 두 가지 속성을 통해 와인 병의 품질을 다른 측면에서 판단하고 관련 산업이 가능한 문제를 개선하는 데 도움이 될 수 있습니다.
다음으로 간단한 회귀분석 모델을 예로 들어 수학 원리를 설명하는데, 여기서는 위키피디아를 직접 활용하여 수학 공식에 대한 간단한 이해를 제공합니다. Simple Linear Regression의 수학적 모델은 다음과 같이 나타낼 수 있습니다.

$y = \beta_0 + \beta_1 x + \varepsilon$

y 종속변수의 값, x 독립변수의 값, β_0 절편(상수항), β_1 인수 x의 계수, ε 오차항

회귀모형에 따라 수학적 표현이 다르기 때문에 여기서는 먼저 더 많은 조사를 하지 않으나, 다른 모델에서 얻은 결과값에 대해서는 통일된 평가방법을 사용할 수 있으며, 여기에서는 본 보고서에 사용된 MSE, MAE

및 R2 값의 수학적 표현입니다.

Mean Squared Error: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Mean Absolute Error: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

Coefficient of Determination: $R^2 = 1 - \frac{SSR}{SST}$

sum of squares for total : $SST = \sum_{i}^{n}(y_i - \bar{y})^2$
sum of squares for regression : $SSR = \sum(\hat{y}_i - \bar{y})^2$

여기서 $\hat{y}$는 모델을 제공하여 얻은 예측 값, y_i는 해당 실제 값, n 은 샘플 수, $\bar{y}$는 y_i의 평균입니다.
회귀모형을 평가하기 위해서는 MSE 와 MAE 의 수치가 작을수록 좋지만 데이터셋에서 목표치인 변수 자체의 수치가 매우 크다면 MSE 와 MAE 의 수치도 상대적으로 커 보일 수 있으므로 훈련 전에 데이터셋을 표준화해야 하며, 이후 얻어진 수치는 0-1 사이이며 0 에 가까울수록 모델의 손실이 적습니다.R2 는 0-1 사이이며 1 에 가까울수록 모델의 적합도가 더 좋습니다.더 많은 선형 회귀 모델에 대한 자세한 설명과 사례는 sklearn 웹 사이트의 설명을 참조하십시오 :
https://scikit-learn.org/stable/modules/linear_model.html
회귀 분석을 분석 도구로 선택한 이유는 scikit-learn 라이브러리를 사용하여 파이썬에서 모델링할 수 있기 때문입니다.scikit-learn 은 머신러닝 모델링을 위한 프로그래밍에 필요한 다양한 함수와 도구를 포함하는 python 기반 머신러닝 라이브러리이며, 데이터 전처리부터 사전 설정 모델 사용, 평가까지 sklearn 을 사용하여 많은 복잡한 프로그래밍 프로세스를 학습하고 수행할 필요 없이 간단하고 쉽게 구현할 수 있습니다.회귀분석 자체는 독립변수와 종속변수의 관계를 설정하고 예측할 수 있다는 특징이 있으며, 다양한 데이터 유형에 유연하게 적용할 수 있고 결과를 이해하기 쉬운 장점이 있습니다.
sklearn 라이브러리에서 사용할 수 있는 다양한 사전 설정 회귀 모델이 제공되었기 때문에 이 보고서에서 단순 회귀 모델, 다중 회귀 모델, 무작위 산림 회귀 모델, 의사 결정 트리 회귀 모델 및 KNN 모델의 5 가지 기본 회귀 모델을 선택하여 데이터 세트를 분석할 계획입니다.마지막으로 이러한 모델의 MSE, MAE 및 R2 값을 비교하여 모델의 훈련 정확도를 평가할 수 있습니다.

II. Datasets

데이터세트 소개: This dataset is related to red variants of spanish wines. The dataset describes several popularity and description metrics their effect on it's quality. The datasets can be used for classification or regression tasks. The classes are ordered and not balanced (i.e. the quality goes from almost 5 to 4 points). The task is to predict either the quality of wine or the prices using the given data.

The dataset contains 7500 different types of red wines from Spain with 11 features that describe their price, rating, and even some flavor description.

데이터세트 섹션:

	winery	wine	year	rating	num_reviews	country	region	price	type	body	acidity
0	Teso La Monja	Tinto	2013	4.9	58	Espana	Toro	995.00	Toro Red	5.0	3.0
1	Artadi	Vina El Pison	2018	4.9	31	Espana	Vino de Espana	313.50	Tempranillo	4.0	2.0
2	Vega Sicilia	Unico	2009	4.8	1793	Espana	Ribera del Duero	324.95	Ribera Del Duero Red	5.0	3.0
3	Vega Sicilia	Unico	1999	4.8	1705	Espana	Ribera del Duero	692.96	Ribera Del Duero Red	5.0	3.0
4	Vega Sicilia	Unico	1996	4.8	1309	Espana	Ribera del Duero	778.06	Ribera Del Duero Red	5.0	3.0

Attributes information:

1.winery: Winery name
2.wine: Name of the wine
3.year: Year in which the grapes were harvested
4.rating: Average rating given to the wine by the users [from 1-5]
5.num_reviews: Number of users that reviewed the wine
6.country: Country of origin [Spain]
7.region: Region of the wine
8.price: Price in euros [€]
9.type: Wine variety
10.body: Body score, defined as the richness and weight of the wine in your mouth [1-5]
11.acidity: Acidity score, defined as wine's “pucker” or tartness; it's what makes a wine refreshing and your tongue salivate and want another sip [1-5]

Source: fedesoriano. (April 2022). Spanish Wine Quality Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/datasets/fedesoriano/spanish-wine-quality-dataset

III. Methodology

1. 프로세스 소개

파이썬을 사용하는 전체 프로세스에는 데이터 가져오기, 전처리, 통계 분석, 훈련 모델 및 결과 표시의 여러 부분이 포함됩니다.모델을 훈련하는 부분에서 우리는 다양한 회귀 모델을 사용했습니다. 선형 회귀, 다항식 회귀, 무작위 포레스트 회귀, 결정 트리 회귀 및 KNN 회귀를 포함합니다. 다음은 간단한 방법의 개요입니다:

1) 데이터 준비: 먼저 csv 파일 데이터 세트를 pandas 데이터 상자로 가져온 다음 데이터 세트의 빈 값을 모두 삭제합니다.그런 다음 비숫자 텍스트 속성을 인코딩한 다음 전체 데이터 세트를 표준화해야 후속 데이터 세트가 훈련 중에 모델에 더 적합할 수 있습니다.
2) 또한 원본 데이터 세트에 포함된 데이터를 분석하여 각 속성 자체와 상호간에 포함된 일부 정보를 얻을 수 있으며 이러한 정보는 통계 계산 및 시각화 방법으로 표시됩니다.
3) 추가 훈련 전에 sklearn 데이터베이스의 도구를 사용하여 데이터 세트를 훈련 세트와 테스트 세트로 나눕니다.그런 다음 가격과 점수를 각각 목표 변수로 사용하여 다음 훈련 과정을 수행합니다.
4) 선형 회귀는 예측을 위해 Linear Regression 클래스의 fit() 방법과 predict() 방법을 호출하여 달성됩니다.원리는 직선을 피팅하여 독립 변수와 종속 변수 간의 선형 관계를 설명하는 것입니다.선형 회귀의 장점은 간단하고 이해하기 쉽고 해석하기 쉽고 계산 효율이 높다는 것입니다.그러나 비선형 관계를 모델링하는 능력은 제한적입니다.
5) 다항식 회귀: 선형 회귀와 유사하지만 x의 제곱 또는 입방체 속성을 포함하도록 예측 변수 x의 행렬을 변환하는 추가 단계를 포함합니다.다항식 회귀는 선형 회귀보다 복잡한 관계에 적합할 수 있으며 상대적으로 유연합니다.단점은 과적합이 쉽고 고차 다항식의 경우 모델의 복잡성이 너무 높을 수 있다는 것입니다.
6) 무작위 숲 회귀: 무작위 숲은 각 트리가 예측을 위해 사용되는 다수의 개별 결정 트리로 구성되며 숲에서 가장 많이 투표된 예측이 최종 예측으로 선택됩니다.높은 견고성과 쉬운 과적합이 없기 때문에 대규모 데이터 세트에 적합합니다.
7) 결정 트리 회귀는 데이터를 다른 트리 구조 경로로 분할하여 연속형 출력에 적합한 연속형 변수를 예측하고 각 잎에 대해 단일 값을 출력하는 데 사용됩니다.결정 트리 회귀는 복잡한 비선형 관계를 포착할 수 있지만 과적합이 쉬울 수 있습니다.
8) KNN 회귀: KNN(K의 가장 가까운 이웃) 모델의 작동 원리는 쿼리와 데이터의 모든 예제 사이의 거리를 찾고 쿼리에 가장 가까운 지정된 수의 예를 선택한 다음 가장 빈번한 레이블을 선택하는 것입니다.KNN 모델도 비선형 문제에 적합하지만 컴퓨터 성능이 높다는 단점이 있습니다.
9) 훈련 결과는 각 모델의 정확도 값을 계산하고 표에 저장하는 데 사용되며, 각 모델에 대해 훈련된 샘플 결과도 원래 데이터 세트와 시각적으로 비교됩니다.

2. 코드 예제

여기에 일부 코드 및 해석을 보여 줄 예정이니, 전체 코드는 첨부파일을 참고하시기 바랍니다.

#데이터 전처리 부분:

```
#import data
data = pd.read_csv('wines_SPA.csv')
#delete the rows with missing values and the column country
data['year'].replace('N.V.', np.nan, inplace=True)
data.dropna(inplace=True)
data.drop(columns=['country'], inplace=True)
#encoding the dataset
from sklearn.preprocessing import LabelEncoder
lable = LabelEncoder()
data['winery'] = lable.fit_transform(data['winery'])
data['wine'] = lable.fit_transform(data['wine'])
```

```
data['region'] = lable.fit_transform(data['region'])
data['type'] = lable.fit_transform(data['type'])
#data scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
data_scaled = pd.DataFrame(data_scaled, columns=data.columns)
#split the dataset, target is price
x_train, x_test, y_train, y_test = train_test_split(data_scaled.drop(columns=['price']),
data_scaled['price'], test_size=0.2, random_state=0)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

#데이터 세트 설명 부분:

```
#check the infomation of the dataset
data.info()
data.describe()
#check the information of the winery
data['winery'].value_counts()
#check the information of the name
data['wine'].value_counts()

#check the distribution of the year
data['year'] = data['year'].astype('int64')
YearCategory = pd.cut(data['year'], bins=[-float('inf'), 2000, 2005, 2010, 2015, 2020,
float('inf')], labels=['<2000', '2000-2004', '2005-2009', '2010-2014', '2015-2019',
'2020+'])
print(YearCategory.value_counts())
#plot the distribution of the year
plt.figure(figsize=(5, 3))
sns.countplot(YearCategory)
plt.title('Year Category')
plt.show()

#check the distribution of the rating
rating = data['rating'].value_counts()
print(rating)
plt.figure(figsize=(5, 3))
plt.hist(data['rating'], bins=20)
plt.title('Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()

#check the distribution of the price
PriceCategory = pd.cut(data['price'], bins=[-float('inf'), 50, 100, 200, 300, 400, 500,
float('inf')], labels=['<50', '50-100', '100-200', '200-300', '300-400', '400-500',
'500+'])
print(PriceCategory.value_counts())
sns.countplot(PriceCategory)
```

```
plt.title('Price Category')
plt.ylabel('Price(EUR)')

#check the relationship between price and rating
plt.figure(figsize=(5, 3))
sns.scatterplot(x='price', y='rating', data=data)
plt.title('Price and Rating')
plt.show()

#check the average price of each rating
price_rating = data.groupby('rating')['price'].mean()
print(price_rating)
plt.figure(figsize=(5, 3))
sns.barplot(x=price_rating.index, y=price_rating.values)
plt.title('Price Rating')
plt.xlabel('Rating')
plt.ylabel('Price(EUR)')
plt.show()

#check the distribution of the type
print(data['type'].value_counts())
plt.figure(figsize=(6, 4))
sns.countplot(data['type'])
plt.title('Type')
plt.show()

#calculate the mean price of each type
mean_price = data.groupby('type')['price'].mean()
print(mean_price)
#plot the mean price of each type
plt.figure(figsize=(6, 4))
sns.barplot(x=mean_price.index, y=mean_price.values)
plt.title('Mean Price')
plt.xlabel('Type')
plt.xticks(rotation=90)
plt.ylabel('Price(EUR)')
plt.show()

#check the seaborn heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(data.corr(), annot=True)
plt.show()
```

#훈련 모형 부분:

```
#defining evaluation function
def evaluate(y_test, y_pred):
    from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
    MSE = mean_squared_error(y_test, y_pred)
    MAE = mean_absolute_error(y_test, y_pred)
    R2 = r2_score(y_test, y_pred)
    print('MSE: ', MSE)
    print('MAE: ', MAE)
    print('R2: ', R2)
    Result = [MSE, MAE, R2]
    return Result
#defining plot function
def plot_result(y_test, y_pred):
    fig, ax = plt.subplots(figsize = (12,4))
    idx = np.asarray([i for i in range(50)])
    width = 0.2
    ax.bar(idx, y_test.values[:50], width = width)
    ax.bar(idx+width, y_pred[:50], width = width)
    ax.legend(["Actual", "Predicted"])
    ax.set_xticks(idx)
    ax.set_xlabel("Index")
    ax.set_ylabel("Price")
    fig.tight_layout()
    plt.show()

    plt.scatter(y_test, y_pred, color='red')
    plt.plot(y_test, y_test, color='blue', linewidth=2)
    plt.xlabel('Actual Price')
    plt.ylabel('Predicted Price')
    plt.show()
#rain the model
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train, y_train)
y_pred = lr.predict(x_test)

#evaluate the model
LR = evaluate(y_test, y_pred)
#plot the results
plot_result(y_test, y_pred)

# compare the results
pd.DataFrame([LR, Poly, RF, DT, KNN], columns=['MSE', 'MAE', 'R2'], index=['Linear
Regression', 'Polynomial Regression', 'Random Forest', 'Decision Tree', 'KNN'])
```

IV. Evaluation & Analysis

Part1: 데이터세트 설명

Table1: 빈 값을 제거한 후의 데이터 세트 정보

```
<class 'pandas.core.frame.DataFrame'>
Index: 6070 entries, 0 to 7499
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   winery       6070 non-null   object
 1   wine         6070 non-null   object
 2   year         6070 non-null   object
 3   rating       6070 non-null   float64
 4   num_reviews  6070 non-null   int64
 5   region       6070 non-null   object
 6   price        6070 non-null   float64
 7   type         6070 non-null   object
 8   body         6070 non-null   float64
 9   acidity      6070 non-null   float64
dtypes: float64(4), int64(1), object(5)
memory usage: 521.6+ KB
```

분석: 모든 null 값을 제거한 데이터 세트는 6,070개의 크기, 9개의 특징 및 특징의 데이터 유형을 가지고 있음을 알 수 있습니다.

Table2: 데이터 집합의 통계

	rating	num_reviews	price	body	acidity
count	6070.000000	6070.000000	6070.000000	6070.000000	6070.000000
mean	4.260115	440.065404	67.397695	4.163756	2.947117
std	0.125122	605.072165	165.514976	0.593981	0.242883
min	4.200000	25.000000	6.260000	2.000000	1.000000
25%	4.200000	388.000000	19.980000	4.000000	3.000000
50%	4.200000	402.000000	31.630000	4.000000	3.000000
75%	4.300000	417.000000	61.940000	5.000000	3.000000
max	4.900000	16505.000000	3119.080000	5.000000	3.000000

분석: 점수 범위는 4.26점으로 평균 4.26점에 불과합니다. 평가 수는 25-16505이며 와인 병당 평균 440개의 평가가 있습니다. 가격은 최저 6.26유로에서 최고 3119유로, 평균 67유로.와인 바디의 평균은 4.16이고 평균 산도는 2.947입니다.

Table3: 생산자수

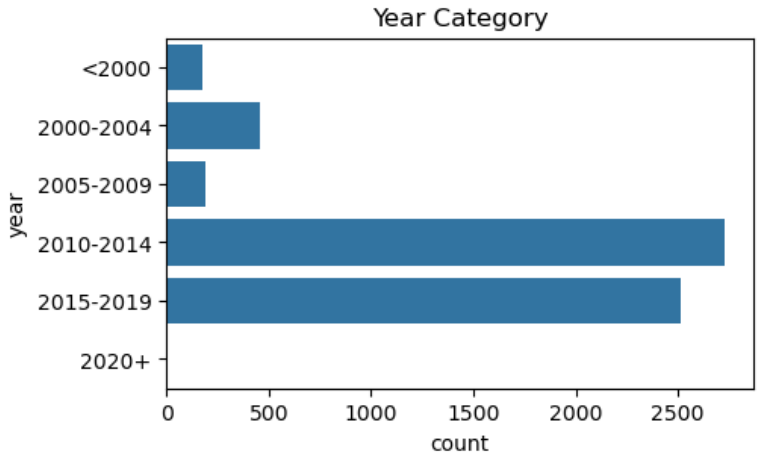
winery	
Contino	414
Artadi	239
La Rioja Alta	228
Sierra Cantabria	215
Vina Pedrosa	207
...	
Particular	1
Bodegas Frontonio	1
Bodegas Asenjo & Manso	1
Micro Bio (MicroBio)	1
Bodegas Monte La Reina	1

Table4: 이름이 나타나는 횟수

wine	
Reserva	422
Gran Reserva	415
Rioja Reserva	218
Valdegines	202
Corimbo I	202
...	
Savinat Sauvignon Blanc	1
Felix Azpilicueta Coleccion Privada	1
Cava Cuvee De Prestige Trepat	1
Solanes Priorat	1
Rioja B70	1

분석: 테이블 3, 4에서 가장 많은 수의 제조업체와 와인 이름을 볼 수 있습니다.생산자는 여러 와이너리를 보유하지 않는 한 지역 및 와이너리가 해당되기 때문에 생산자는 지역 및 유형과 일정한 관계가 있을 수 있습니다.와인 이름은 또한 가격과 일정한 관계가 있을 수 있는데, 표의 상위 순위에서 등급 및 숙성 시간과 관련된 명명 방법인 Reserva가 나타나는 것을 볼 수 있는데, 예를 들어 1위 Reserva는 이 와인 병을 나무 통에 최소 3년 동안 보관해야 하는 반면 2위 Gran Reserva는 최소 5년 동안 보관해야 함을 나타냅니다.그러면 시간 비용으로 인해 숙성된 레드 와인의 가격은 일반적으로 짧거나 숙성되지 않은 레드 와인보다 높습니다.

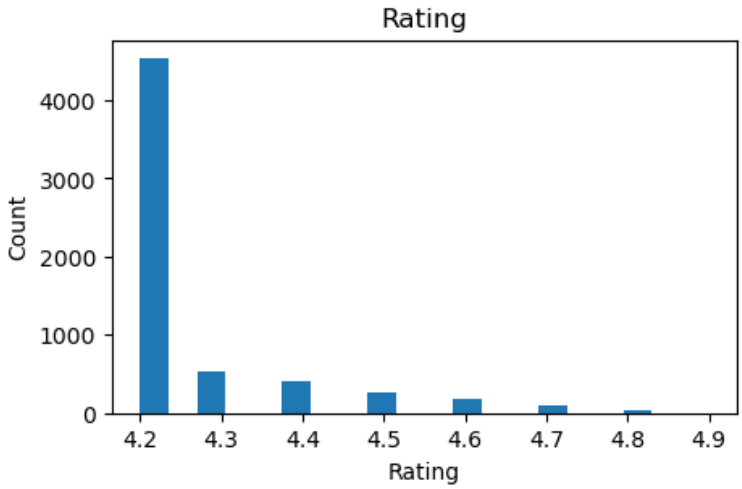
Fig1: 연도분포



year	
2010-2014	2729
2015-2019	2510
2000-2004	459
2005-2009	192
<2000	175
2020+	5

분석: 우리가 인용한 데이터 세트의 출시 시기는 2022년으로 대부분의 레드 와인이 2010-2019년임을 알 수 있으며 레드 와인의 보존은 비교적 높은 조건이 필요하기 때문에 더 일찍 생산된 레드 와인의 수는 지난 10년 레드 와인보다 훨씬 적습니다.

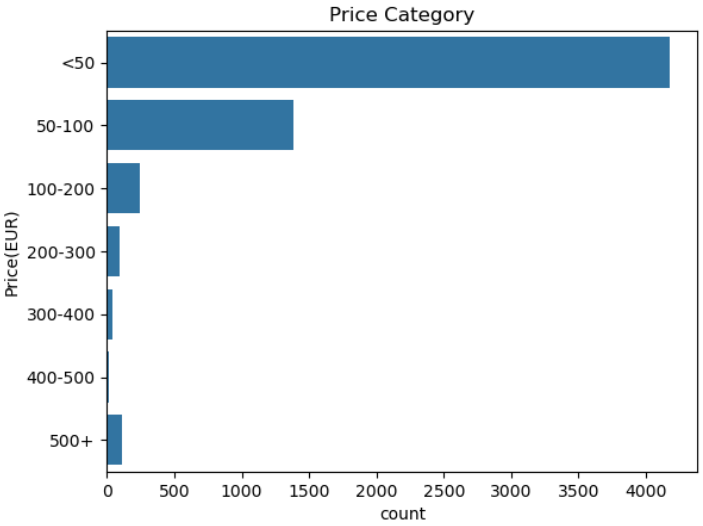
Fig2: 채점분포



rating	
4.2	4544
4.3	528
4.4	402
4.5	264
4.6	182
4.7	105
4.8	43
4.9	2

분석: 대부분의 점수가 4.2점임을 알 수 있으며 이는 표에서 가장 낮은 수치이지만 데이터셋 소개와 결합하여 채점 기준은 1-5점이므로 4.2점은 이미 높은 점수를 받은 와인에 속하므로 데이터셋에 포함된 모든 와인의 품질은 나쁘지 않을 것입니다.

Fig3: 가격분포



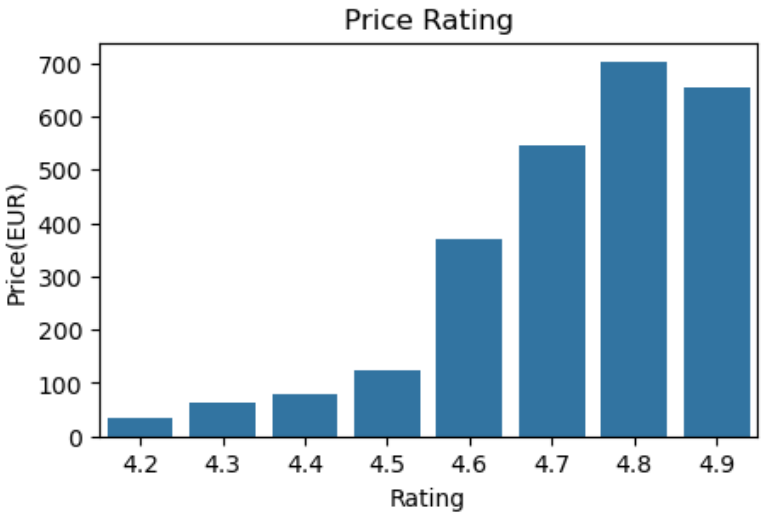
price	
<50	4175
50-100	1385
100-200	245
500+	111
200-300	92
300-400	45
400-500	17

분석: 데이터 세트에서 대부분의 레드 와인의 가격은 50유로를 초과하지 않으며 100유로를 초과하는 레드 와인은 극소수에 불과함을 알 수 있습니다.

Fig4: 평점의 가격분포

분석: 이 그림에서 각 점수의 각 판매 가격 분포를 볼 수 있으며 4.2점은 모두 매우 저렴한 레드 와인이며 점수가 높을수록 가격 범위가 점차 증가합니다.

Fig5: 평점평균가격



rating	
4.2	34.340713
4.3	61.797759
4.4	78.764162
4.5	124.602754
4.6	369.175754
4.7	544.281334
4.8	702.889474
4.9	654.250000

분석: 각 점수의 평균 가격을 관찰하면 4.2~4.8점 사이에서 점수가 증가함에 따라 평균 가격도 선형적으로 증가함을 알 수 있으며, 4.9점은 3개의 샘플만 있기 때문에 평균 가격의 참고 의미는 크지 않습니다.

Fig6: 품종의 분포 상황

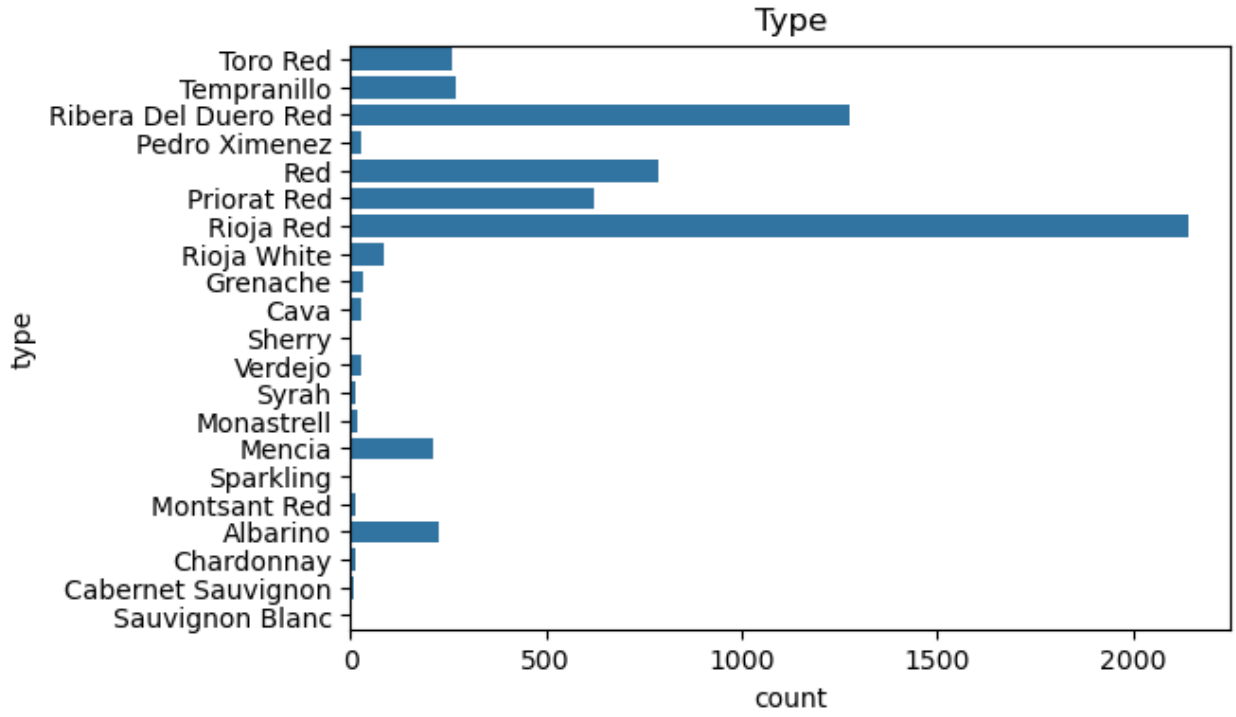


Fig6-table:

type			
Rioja Red	2143	Pedro Ximenez	27
Ribera Del Duero Red	1277	Verdejo	26
Red	786	Monastrell	17
Priorat Red	620	Montsant Red	14
Tempranillo	267	Syrah	13
Toro Red	261	Chardonnay	13
Albarino	225	Cabernet Sauvignon	10
Mencia	213	Sparkling	5
Rioja White	86	Sauvignon Blanc	3
Grenache	33	Sherry	2
Cava	29		

분석: 품종분포에서는 거의 모두 레드와인이 상위권을 차지하고 있으며, 이는 레드와인이 와인의 주요 카테고리임을 보여준다. 셰리, PX, 카바 등 가장 적게 등장하는 품종은 모두 스페인을 대표하는 와인이고, 그밖에 덜 등장하는 품종은 주로 화이트 와인, 스파클링 와인, 스위트 와인인데 왜 맨 마지막에 순위를 매겼을까? 레드와인은 포도 자체의 맛에 더 중점을 두기 때문에 보통 수확연도와 생산연도를 표시하는데 반해 이들 후기 와인 종류는 대부분 포도맛에 중점을 둡니다. 포도 자체의 맛에 대한 기준이 높지 않은 와인을 사용하여 만들어지며, 양조 과정에서 만들어지는 맛에 중점을 둔 와인의 경우 대부분 연도가 표기되지 않고 데이터 세트 전처리 과정에서 삭제됩니다.

Fig7: 품종평균가격

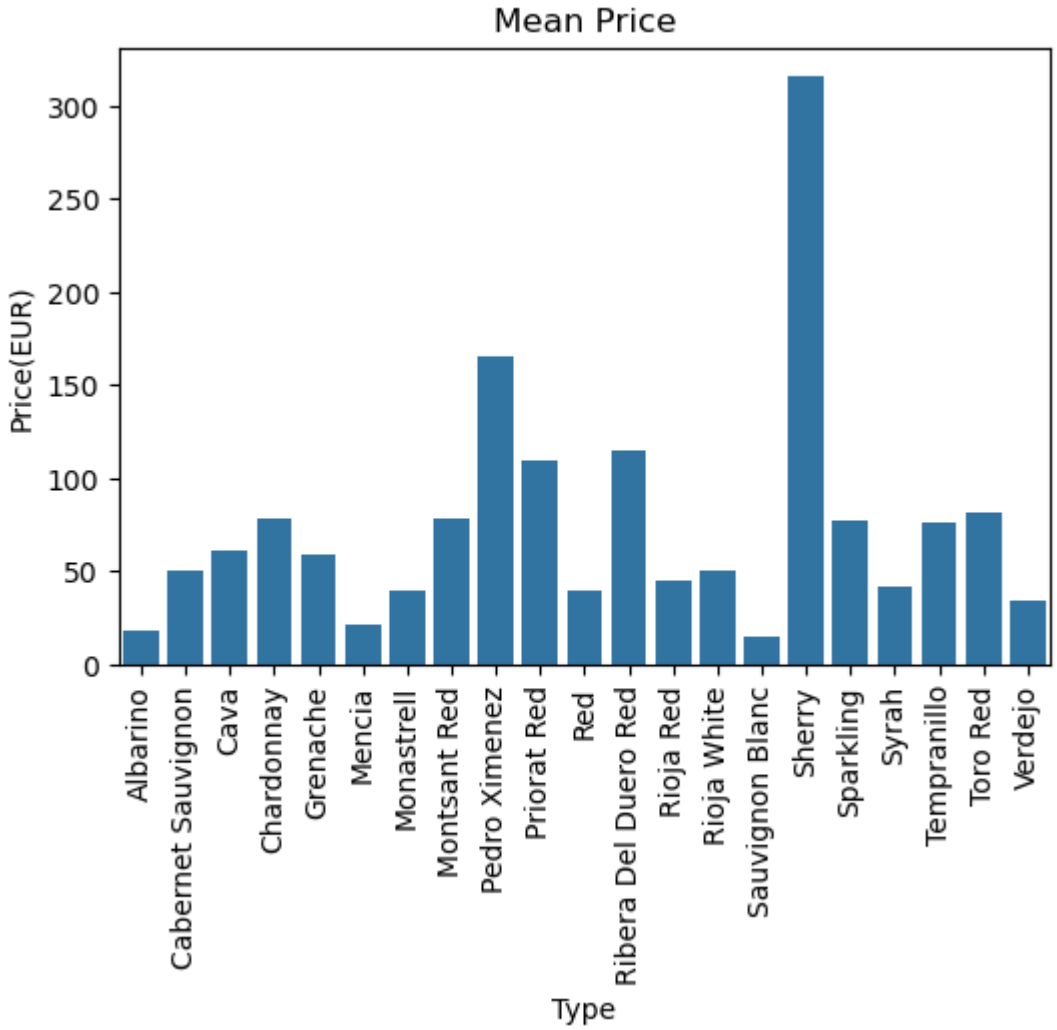


Fig7-table:

type	
Albarino	18.054167
Cabernet Sauvignon	49.866000
Cava	61.354138
Chardonnay	78.114615
Grenache	58.977879
Mencia	21.458122
Monastrell	38.924118
Montsant Red	77.940000
Pedro Ximenez	165.625926
Priorat Red	109.540832
Red	38.835062
Ribera Del Duero Red	115.095126
Rioja Red	44.556051
Rioja White	50.220581
Sauvignon Blanc	14.333333
Sherry	315.425000
Sparkling	76.794000
Syrah	41.903160
Tempranillo	75.461738
Toro Red	81.049923
Verdejo	33.968077

분석: 품목별 평균 가격을 살펴보면 큰 차이가 없다는 결론을 내렸습니다.그 중 셰리와 PX 의 평균 가격이 다른 종류보다 훨씬 높은 이유는 샘플 수가 너무 적기 때문이며, 앞서 언급한 정보와 결합하여 셰리는 일반적으로 연도를 표시하지 않기 때문에 대부분 전처리에서 삭제된 반면, 연도가 남아 있는 셰리는 매우 비싸고 희귀한 제품입니다.

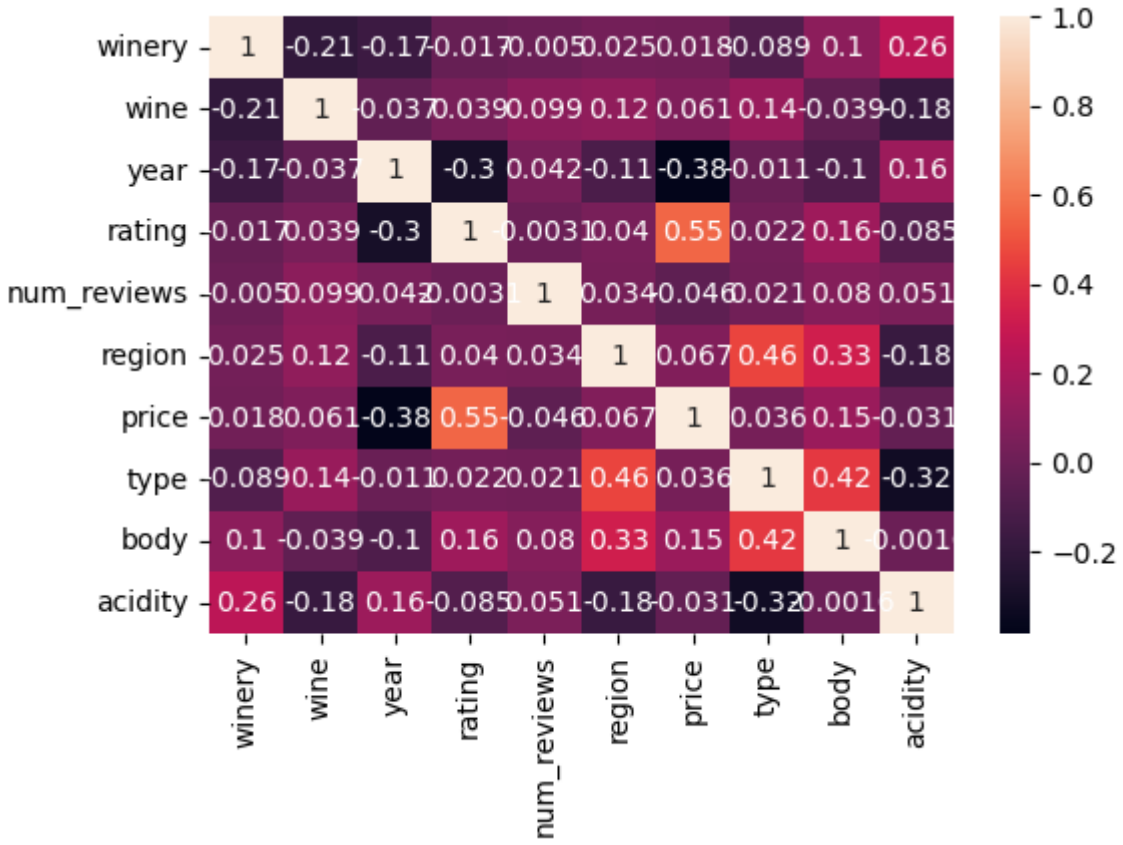
Table5, 6: 주체와 산도의 분포 상황

body	
4.0	3868
5.0	1615
3.0	553
2.0	34

acidity	
3.0	5776
2.0	267
1.0	27

분석: 데이터 세트에서 대부분의 레드 와인의 바디가 편중되어 있음을 알 수 있습니다.데이터 세트 소개에서 산도 범위가 1~5점이라고 언급했기 때문에 대부분의 산도는 3점입니다.

Fig8: 히트맵



Seaborn 의 히트 맵을 사용하여 행렬의 다양한 요소 간의 상관 관계를 표시할 수 있습니다. 상관관계의 강도와 방향은 색상의 깊이나 양의 색과 음의 색으로 표현됩니다. 먼저 상관관계가 높은 속성을 관찰했는데, 그 중 가격과 점수 사이의 상관관계가 가장 높았으며, 이는 일반적으로 점수가 높은 레드 와인의 가격이 더 높다는 것을 나타냅니다. 다음은 지역과 종류인데, 일부 레드 와인의 경우 산지 보호를 받기 때문에 이들 레드 와인의 산지는 반드시 동일해야 합니다. 그러면 우리는 생산지, 종류, 와인 바디의 세 가지 속성이 서로 일정한 상관관계를 가지고 있음을 알 수 있습니다. 이러한 상관관계가 나타나는 이유는 이전 글에서 언급한 지정된 산지와 결합하여 특정 유형의 레드 와인을 생성할 수 있으며, 이러한 특정 유형의 레드 와인은 종종 독특한 특징을 가지고 있습니다.예를 들어 페드로 히메네스(Pedro Ximénez)는 설탕이 함유된 슈퍼 와인입니다. 스위트 와인은 설탕이 너무 많아 점도가 매우 높기 때문에 와인 바디 점수가 가장 높을 것입니다. 남부 안달루시아 지역에서 생산되기 때문에 와인 바디와 생산 지역 사이에는 일정한 상관관계가 있습니다. 그러면 히트맵에서 산도는 종류와 음의 상관관계가 있고, 지역과도 음의 상관관계가 있지만 와인 바디와는 거의 상관관계가 없다는 것을 알 수 있습니다. 이는 종류가 와인 바디와 산도 모두에 영향을 미친다는 것을 의미합니다. 어떤 영향은 있지만 와인 바디와 산도 사이에는 영향이 없습니다.

Part2: 훈련 결과

요약: 각 모델의 정확도는 R2 점수, 평균 절대 오차(MAE) 및 평균 제곱 오차(MSE)를 사용하여 평가됩니다. 이 모든 작업은 평가 기능에서 수행됩니다. 이러한 측정항목은 각각 모델 예측 정확도, 편향 및 오류 크기에 대한 중요한 정보를 제공합니다. 마지막으로, 이러한 평가 결과를 시각화하려면 플롯_결과 함수의 플롯을 사용하십시오.

Table1: 가격을 목적함수로 사용했을 때 각 모델이 구하는 정확도 값은

	MSE	MAE	R2
Linear Regression	0.711549	0.303997	0.365149
Polynomial Regression	0.489325	0.239042	0.563420
Random Forest	0.455591	0.134605	0.593517
Decision Tree	0.506545	0.132651	0.548055
KNN	0.420998	0.129632	0.624382

가격을 목적함수로 사용하면 KNN이 5개 모델 중 정확도가 가장 높고 손실이 가장 낮습니다. Random Forest 다음으로 정확도가 KNN보다 약간 낮습니다.

Table2: 점수를 목적함수로 사용할 경우, 각 모델에서 얻은 정확도 값은

	MSE	MAE	R2
Linear Regression	0.710216	0.584296	0.330860
Polynomial Regression	0.488027	0.430629	0.540198
Random Forest	0.185284	0.172695	0.825432
Decision Tree	0.279959	0.186982	0.736233
KNN	0.259288	0.200150	0.755708

점수를 목적함수로 사용하면 랜덤 포레스트는 정확도와 손실이 가장 낮은 모델이 되며, 가격을 사용했을 때보다 정확도는 어느 정도 향상되고 손실은 어느 정도 감소합니다. 목표. 두 번째로 적합한 모델은 KNN입니다.

훈련 결과 비교 차트:

첫 번째는 가격을 목표로 삼은 경우,
a. 예측값과 실제값의 비교 파란색은 실제값, 노란색은 모델이 예측한 값입니다. 관찰의 편의를 위해 이 그림에는 데이터 세트의 처음 50개 샘플에 대한 비교만 표시됩니다.
b. 피팅 곡선, 빨간색 점은 예측 가격을 나타내며, 중간 기울기에 가까울수록 실제 가격에 가까우며, 모든 빨간색 점의 분포가 기울기에 가까울수록 모델의 피팅 정도가 좋은 것입니다.
이 부분은 위의 정확도 값과 연동하여 볼 수 있습니다.

Fig1，a：linear regression model

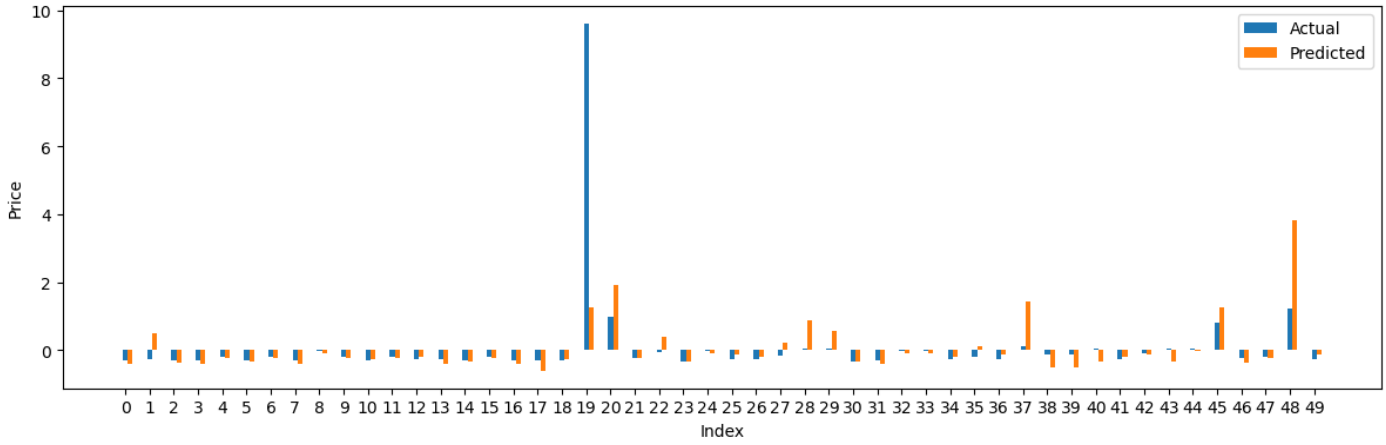


Fig1，b：

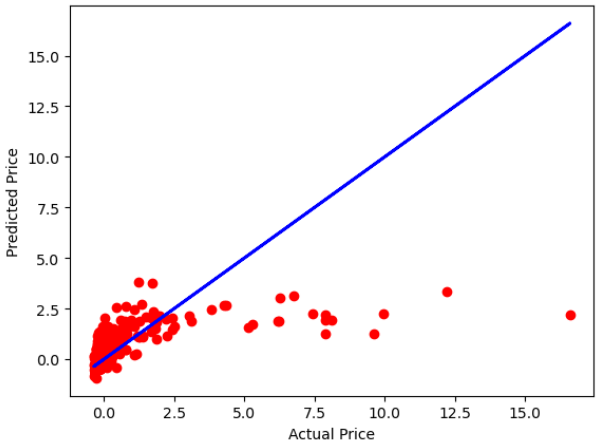


Fig2，a：Polynomial regression model

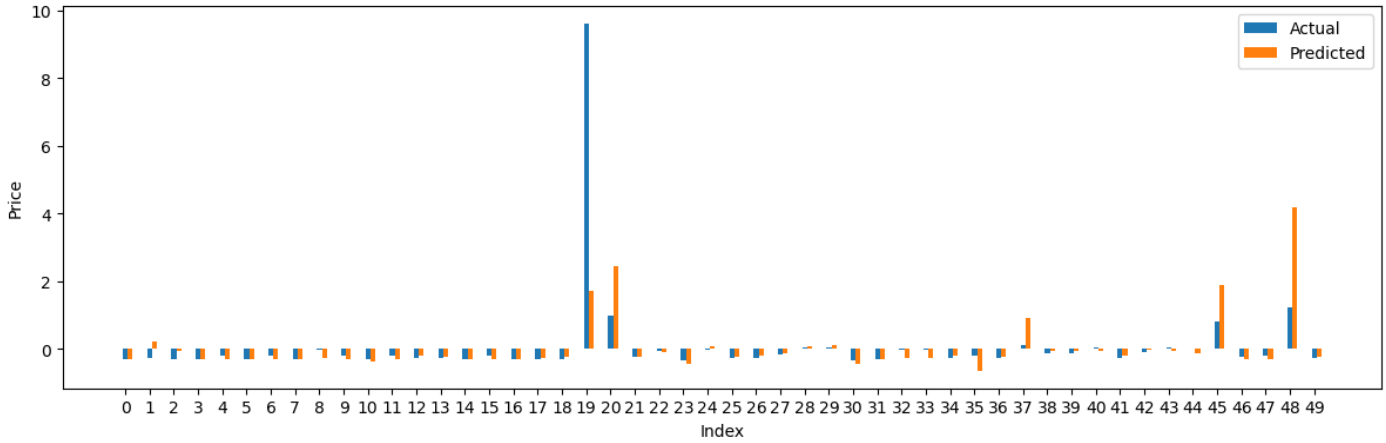


Fig2，b：

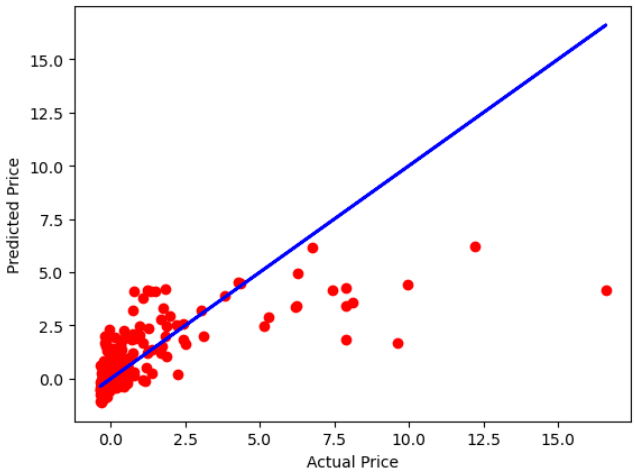


Fig3，a：Random forest regression model

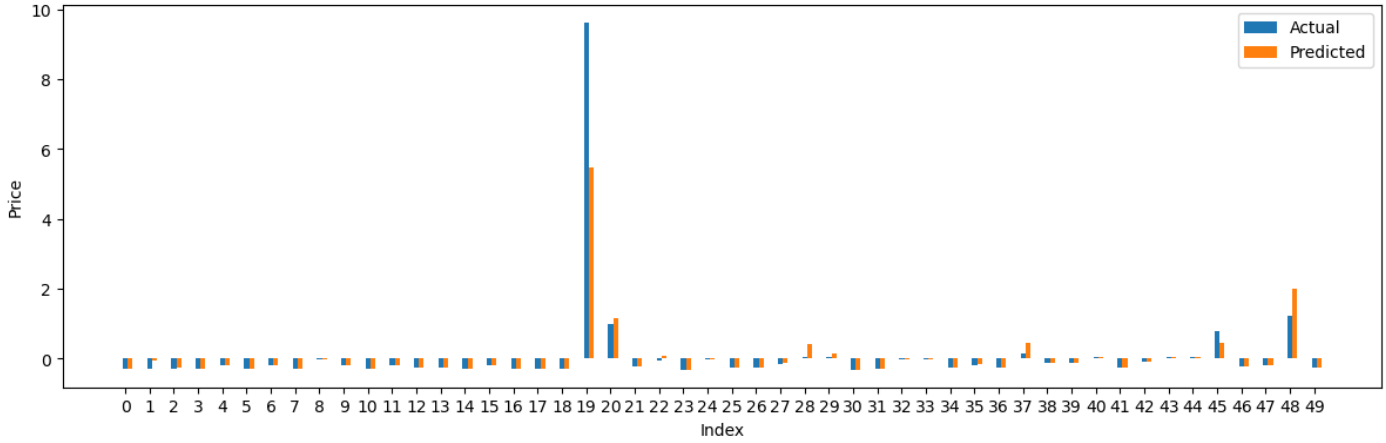


Fig3，b：

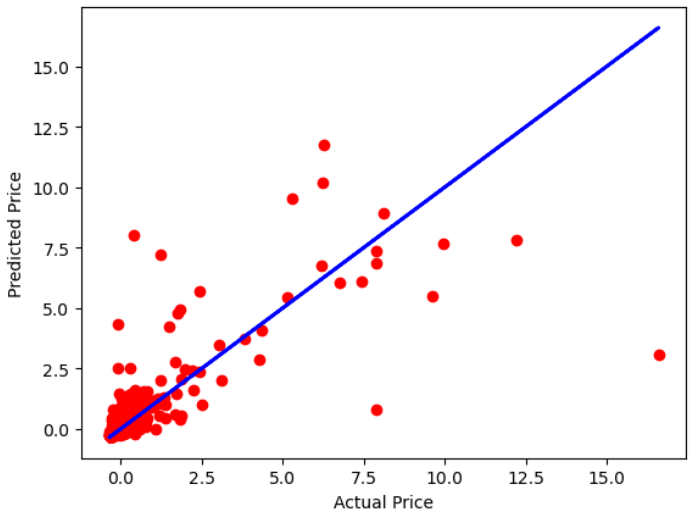


Fig4，a：Decision tree regression model

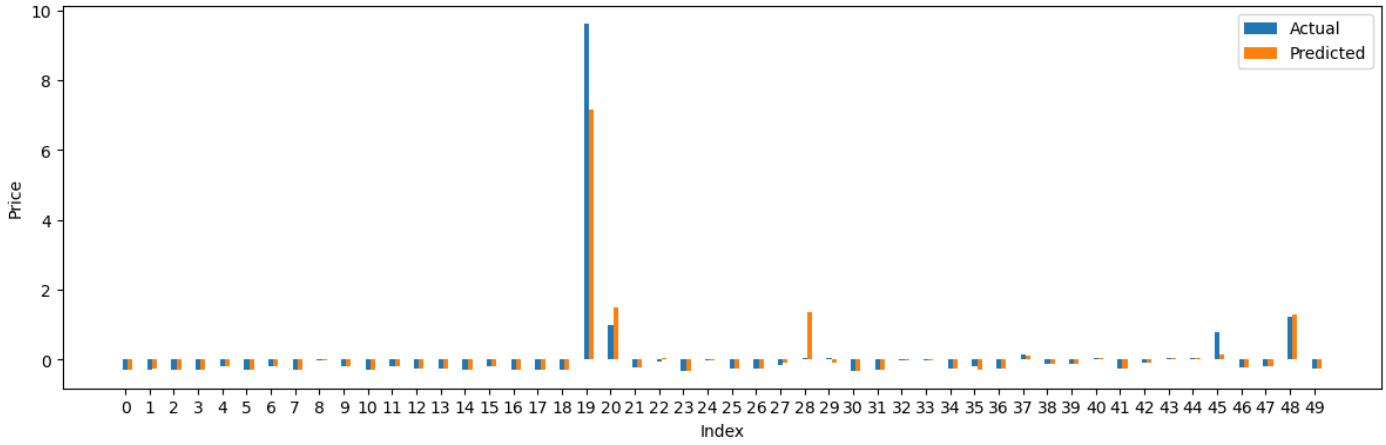


Fig4，b：

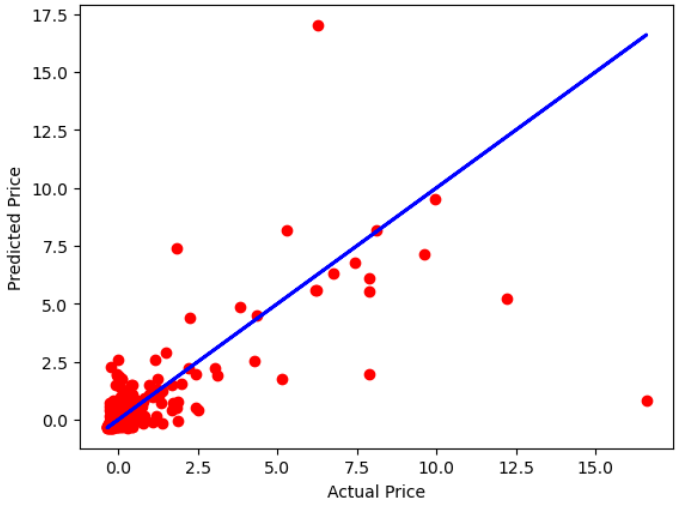


Fig5，a：KNN model

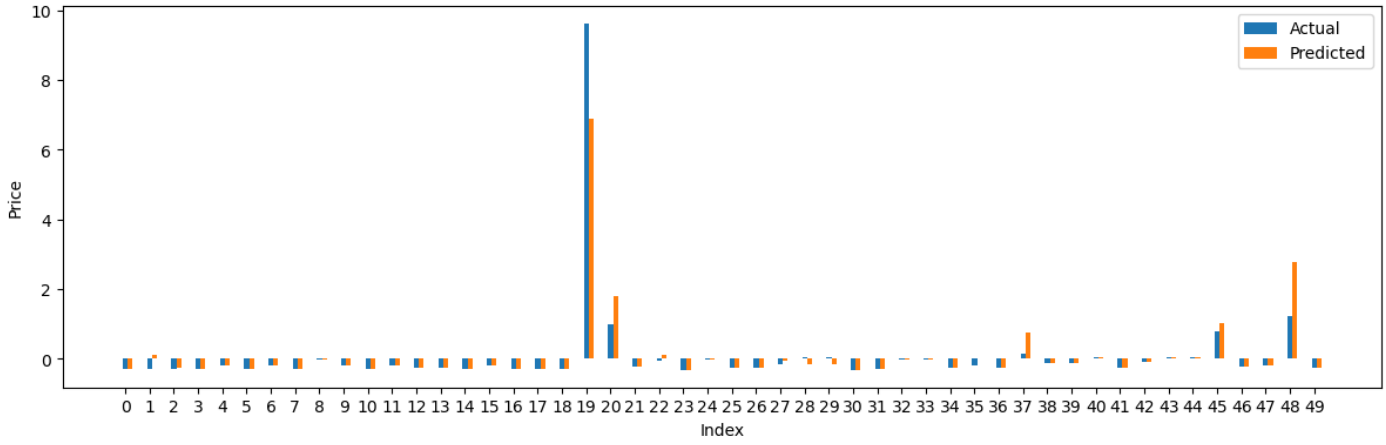
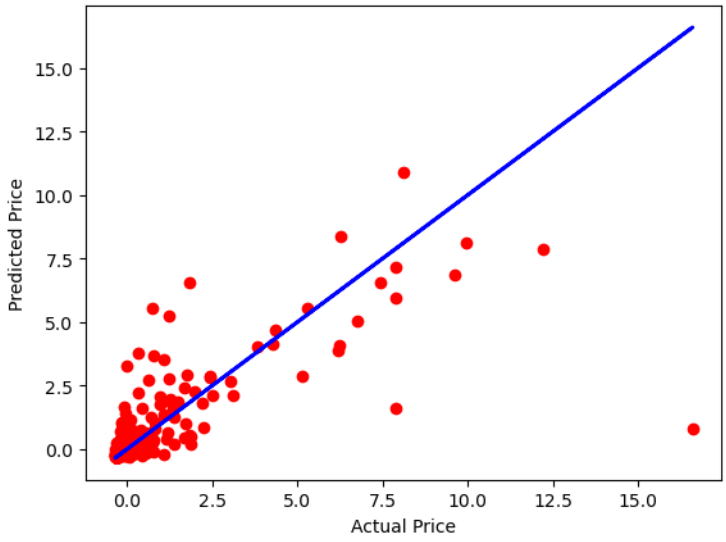


Fig5，b：



채점을 목표로 하는 경우:

이 부분에서는 적합도 곡선 그래프를 생략하는데, 점수 분포가 매우 가깝기 때문에 적합도 곡선 그래프의 예측 값이 매우 정렬된 행렬을 나타내어 우리의 관찰에 도움이 되지 않습니다.

Fig6: linear regression model

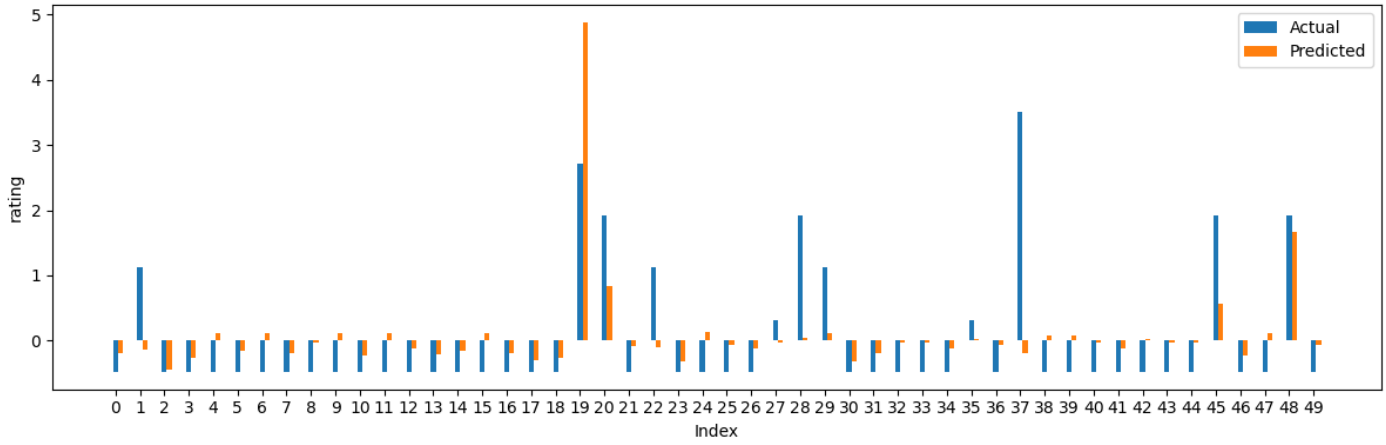


Fig7: Polynomial regression model

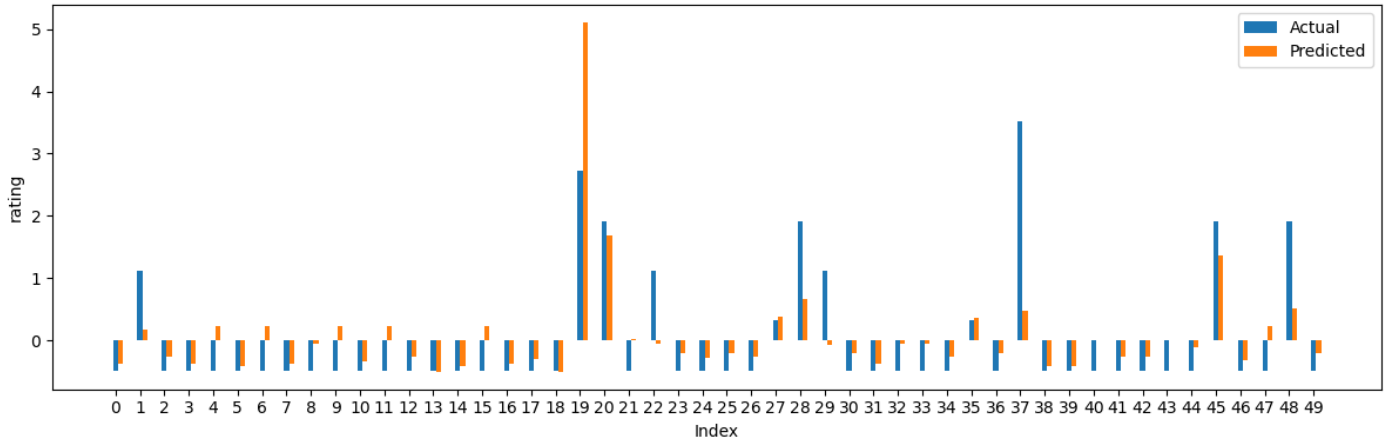


Fig8：Random forest regression model

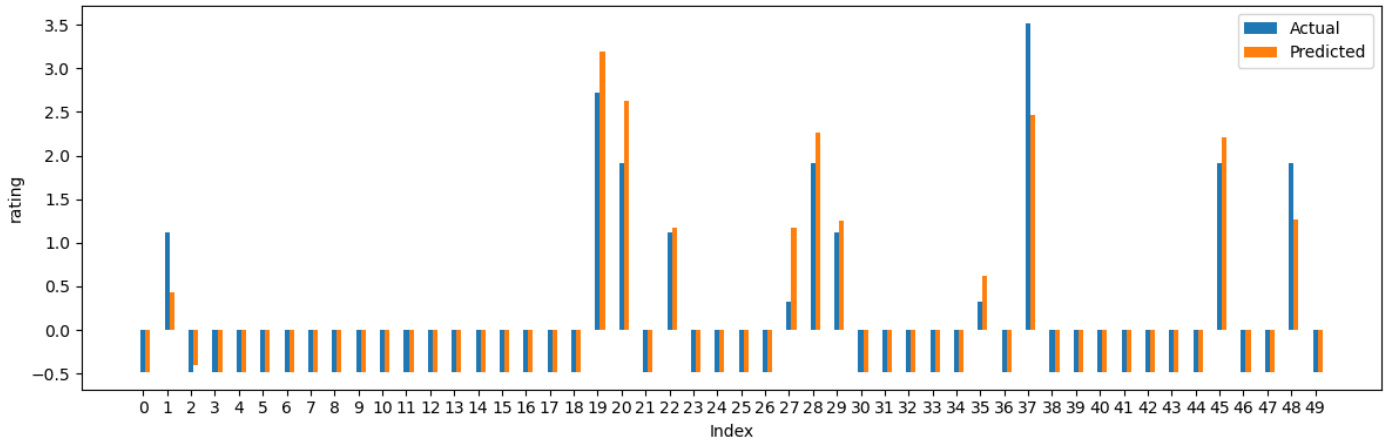


Fig9：Decision tree regression model

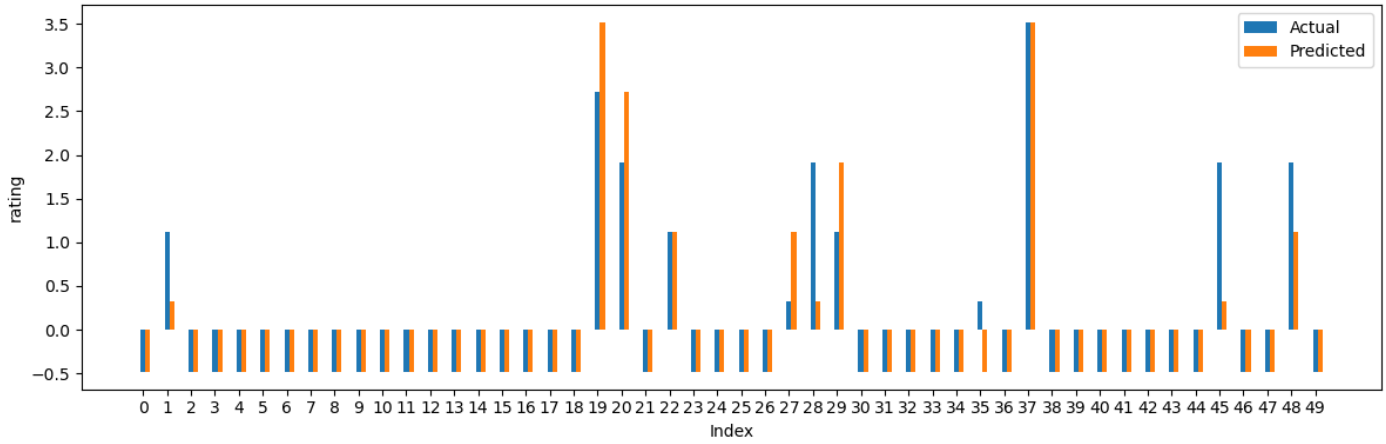
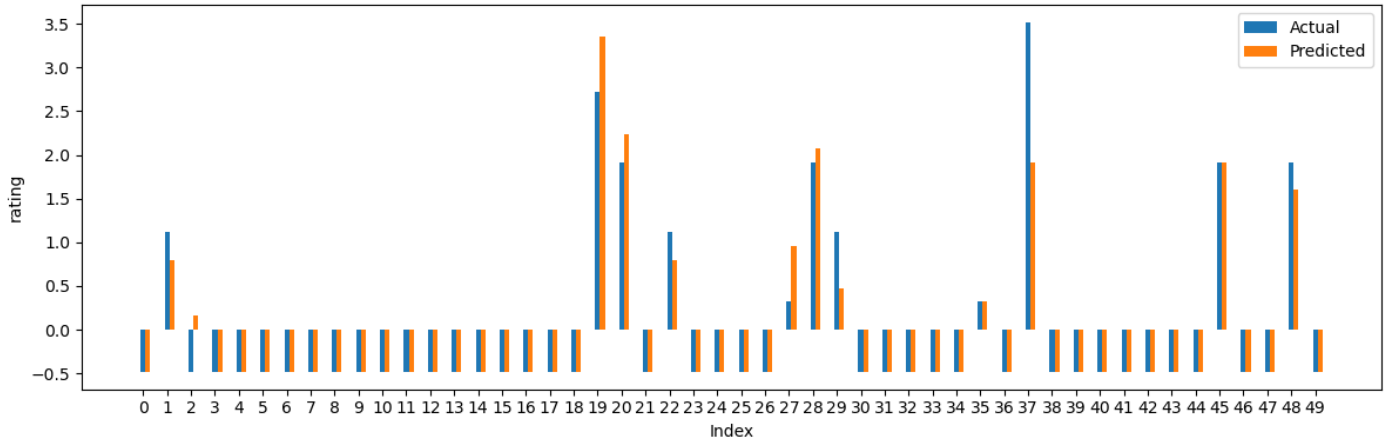


Fig10：KNN model



V. Conclusion: Discussion

스페인 레드와인의 품질을 예측하는 프로젝트에서 우리는 파이썬의 여러 회귀 모델을 사용하여 스페인 레드와인의 등급 및 가격 책정 데이터를 분석했습니다. 목적은 다양한 통계 모델을 사용하여 이러한 변수를 훈련하고 예측하는 것입니다. 우리의 연구는 Kaggle에 제공된 스페인 와인 품질과 관련된 데이터 세트를 사용하여 시작되었으며, 속성 분석 및 대상 변수 사용 모델 훈련을 예측하여 강력한 결과와 풍부한 데이터 통찰력을 얻었습니다.우리의 프로젝트는 회귀 모델을 사용하여 스페인 레드와인의 품질을 정확하게 예측할 수 있음을 보여줍니다. 단순 회귀, 다중 회귀, 무작위 산림 회귀, 의사 결정 트리 회귀 및 KNN 모델을 테스트한 후 이러한 모델의 휴리스틱 능력이 다양하며 그 중 일부는 다른 모델보다 더 정확한 결과를 나타냅니다.훈련 결과는 등급을 대상 변수로 사용할 때 무작위 숲 모델이 가장 높은 정확도와 최소 손실을 얻을 수 있음을 보여주었습니다.따라서 점수를 대상 변수로 사용하고 무작위 포레스트 모델을 사용하는 것이 이 데이터 세트에 가장 적합합니다.초점은 일부 와인 평가 등급이 와인의 전체 품질에 상당한 상관 관계가 있음을 보여주는 연구입니다. 이 결과는 와인의 품질이 와인 등급에 영향을 미치는 중요한 결정 요인임을 증명하며, 등급이 높은 와인은 품질이 우수하고 와인 생산자의 경우 더 높은 가격으로 판매할 수 있습니다. 이러한 흥미로운 발견에도 불구하고 우리 모델의 예측은 모델의 정확도를 향상시키기 위해 데이터 세트를 세분화하는 이점을 얻을 수 있습니다. 예를 들어, 와인의 다른 변수 또는 특성을 분석에 통합하면 평가에 더 많은 복잡성과 진정성을 추가하는 동시에 등급 범위를 확장하고 나중에 모델 교육을 위해 더 완전한 데이터 세트를 사용할 때 더 높은 정확도를 얻을 수 있습니다.그러나 우리의 결과는 한계가 없는 것이 아니며 그 중 하나는 데이터 세트가 스페인 와인에만 집중되어 있다는 것입니다. 이것은 이 와인 유형에 대한 구체적인 이해를 제공할 수 있지만 연구 결과는 다른 유형의 와인이나 다른 지리적 지역의 와인에 일반적으로 적용되지 않을 수 있습니다. 또한 다른 측정되지 않은 요인이 와인 등급이나 가격에 영향을 미칠 가능성도 고려되지 않았습니다.
결론, 회귀 모델을 사용하여 스페인 와인의 품질을 정확하게 예측하는 것은 통찰력 있는 관점을 제공합니다.이러한 발견은 양조업자, 와인 마케터 및 소비자에게 큰 영향을 미칠 수 있습니다.앞으로도 예측 모델의 정확성을 지속적으로 개선하고 개선하고 데이터 세트를 확장하여 더 넓은 범위의 와인과 변수를 포함할 수 있습니다.이러한 데이터 세트에는 여러 국가의 와인이 포함되거나 다른 유형의 와인이 포함될 수 있습니다. 이것은 우리의 발견을 보다 보편적으로 만들고 와인 생산 및 소비의 증가하는 이질성을 충족시킬 수 있습니다.

VI. Related Work (e.g., existing studies)

이 프로젝트에는 다음 출처의 정보가 사용되었습니다.

스페인 레드 와인에 대한 기본 소개: https://en.wikipedia.org/wiki/Spanish_wine
스페인 레드 와인에 대한 기본 소개: https://www.thewinesociety.com/discover/explore/regional-guides/spanish-wine-ultimate-guide
scikit-learn 사용 방법 및 모델 소개: https://scikit-learn.org/stable/user_guide.html
선형 회귀 소개: https://en.wikipedia.org/wiki/Linear_regression
데이터 세트 소스: https://www.kaggle.com/datasets/fedesoriano/spanish-wine-quality-dataset