Виконала студентка ІП-13 Лисенко Анастасія

Лабораторна робота №5

1. Дослідити дані, підготувати їх для побудови регресійної моделі
2. Розділити дані на навчальну та тестову вибірки
3. Побудувати декілька регресійних моделей для прогнозу якості вина (12 - quality). Використати лінійну одномірну та багатомірну регресію та поліноміальну регресію обраного вами виду (3-5 моделей)
4. Використовуючи тестову вибірку, з'ясувати яка з моделей краща

Встановлюємо потрібні бібліотеки

In [1]:
```
!pip install pandas -q
```

```
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
You should consider upgrading via the 'D:\DA\LAB_5\venv\Scripts\python.exe -m pip install --upgrade pip' comm
and.
```

In [2]:
```
!pip install scikit-learn -q
```

```
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
You should consider upgrading via the 'D:\DA\LAB_5\venv\Scripts\python.exe -m pip install --upgrade pip' comm
and.
```

In [3]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
```

Зчитуємо файл

In [4]:
```
path = 'data/winequality-red.csv'

dataset = pd.read_csv(path, sep=',', decimal='.')
```

Аналізуємо дані

In [5]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

In [6]:
```
dataset.head()
```

Out[6]:

	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

In [7]:
```
dataset.describe()
```

Out[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.0
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.3
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.1
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.7
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.2
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.3
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.4
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.0

Очищаємо дані: заміна пустих значень на середні

In [8]:
```
def data_transformation(dataset, columns):
    dataset.fillna(dataset.mean(numeric_only=True), inplace=True)
    for column in columns:
        dataset[column] = dataset[column].abs()

data_transformation(dataset, dataset.columns.to_list())
```

Бачимо, що параметр з найбільшим кофіцієнтом кореляції до quality є alcohol (0.476166)

In [9]:
```
dataset.corr()
```

Out[9]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397

За допомогою функції train_test_split ділимо основні дані на навчальну та тестову

In [10]:
```
wine_train_selection, wine_test_selection = train_test_split(dataset)
```

Розділяємо дані по колонці quality

In [11]:
```
def division(dataset, column):
    x_columns = dataset.loc[:, dataset.columns != column]
    y_column = dataset[column]
    return x_columns, y_column

x_wine_train_sel, y_wine_train_sel = division(wine_train_selection, "quality")
x_wine_test_sel, y_wine_test_sel = division(wine_test_selection, "quality")
```

Виділяємо ознаку alcohol для наших виборок

In [12]:
```
x_alcohol_train = x_wine_train_sel[["alcohol"]]
x_alcohol_test = x_wine_test_sel[["alcohol"]]
```

Будуємо лінійну регресію за ознакою alcohol

In [13]:
```
from sklearn.linear_model import LinearRegression

linear_regression = LinearRegression().fit(x_alcohol_train, y_wine_train_sel)
```

Будуємо багатовимірну регресію за всіма ознаками

In [14]:
```
multivarative_regression = LinearRegression().fit(x_wine_train_sel, y_wine_train_sel)
```

Будуємо поліноміальну регресію другого ступеню за ознакою alcohol

In [15]:
```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import PolynomialFeatures

polynomial_regression = make_pipeline(PolynomialFeatures(degree=2), LinearRegression()).fit(x_wine_train_sel
```

In [19]:
```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
def test_regression(regression, x_test, y_test):
    predictions = regression.predict(x_test)
    mean_squared_erorr_result = mean_squared_error(y_test, predictions)
    mean_absolute_erorr_result = mean_absolute_error(y_test, predictions)
    r2_result = r2_score(y_test, predictions)
    return mean_squared_erorr_result, r2_result, mean_absolute_erorr_result
```

Знаходимо MSE, R2 та MAE для наших регресій

In [20]:
```
linear_mean_squared, linear_r2, linear_mean_absolute = test_regression(linear_regression, x_alcohol_train, y
multivarative_mean_squared, multivarative_r2, multivarative_mean_absolute = test_regression(multivarative_re
polynomial_mean_squared, polynomial_r2, polynomial_mean_absolute = test_regression(polynomial_regression, x_
```

Виводимо результати досліджень

In [21]:
```
print("Linear regression mean squared error: ", linear_mean_squared)
print("Linear regression r2 error: ", linear_r2)
print("Linear regression mean absolute error: ", linear_mean_absolute)

print("Multivarative regression mean squared error: ", multivarative_mean_squared)
print("Multivarative regression r2 error: ", multivarative_r2)
print("Multivarative regression mean absolute error: ", multivarative_mean_absolute)

print("Polynomial regression mean squared error: ", polynomial_mean_squared)
print("Polynomial regression r2 error: ", polynomial_r2)
print("Polynomial regression mean absolute error: ", polynomial_mean_absolute)
```

```
Linear regression mean squared error:  0.4743922876778837
Linear regression r2 error:  0.23142904312418477
Linear regression mean absolute error:  0.5468550705245288
Multivarative regression mean squared error:  0.4031525908818852
Multivarative regression r2 error:  0.3668456790101018204
Multivarative regression mean absolute error:  0.49486850029723
Polynomial regression mean squared error:  0.3496730481278221 3
Polynomial regression r2 error:  0.43348878939666546
Polynomial regression mean absolute error:  0.4603040199652239 7
```

У цьому випадку ми бачимо, що поліноміальна регресія має найнижчу MSE і найвище значення R2, що вказує на те, що вона може бути найкращою моделлю.

In [ ]: