Контрольна робота №1 та №2 (ОДЗ)

In [7]:
```
import pandas as pd
import numpy as np
from sklearn.metrics.regression import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.linear_model import LinearRegression, LassoCV, Lasso
from sklearn.ensemble import RandomForestRegressor
```

У завданні буде використано набір даних про якість білого вина(репозиторій UCI) archive.ics.uci.edu/ml/machine-learning-databases/wine-quality. Завантажте дані

In [2]:
```
!mkdir -p data
!wget -P data -c https://archive.ics.uci.edu/ml/machine-learning-databases/wine-

data = pd.read_csv("data/winequality-white.csv", sep=";")
display(data.sample(10))
```

```
--2020-09-20 15:34:35--  https://archive.ics.uci.edu/ml/machine-learning-databa
ses/wine-quality/winequality-white.csv (https://archive.ics.uci.edu/ml/machine-
learning-databases/wine-quality/winequality-white.csv)
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443...
 connected.
HTTP request sent, awaiting response... 200 OK
Length: 264426 (258K) [application/x-httpd-php]
Saving to: ‘data/winequality-white.csv’

winequality-white.c 100%[===================>] 258.23K  1.03MB/s    in 0.2s

2020-09-20 15:34:36 (1.03 MB/s) - ‘data/winequality-white.csv’ saved [264426/26
4426]
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
901	8.0	0.29	0.29	13.2	0.046	26.0	113.0	0.99830	3.25	0.37	9
144	8.1	0.20	0.40	2.0	0.037	19.0	87.0	0.99210	3.12	0.54	11
4238	6.4	0.29	0.18	15.0	0.040	21.0	116.0	0.99736	3.14	0.50	9
2201	6.0	0.28	0.27	2.3	0.051	23.0	147.0	0.99400	3.23	0.67	10
2544	6.9	0.32	0.30	1.8	0.036	28.0	117.0	0.99269	3.24	0.48	11
3330	6.7	0.23	0.33	8.1	0.048	45.0	176.0	0.99472	3.11	0.52	10
3070	6.8	0.28	0.43	7.6	0.030	30.0	110.0	0.99164	3.08	0.59	12
401	6.8	0.37	0.51	11.8	0.044	62.0	163.0	0.99760	3.19	0.44	8
695	6.3	0.34	0.28	14.7	0.047	49.0	198.0	0.99770	3.23	0.46	9
1778	6.4	0.15	0.36	1.8	0.034	43.0	150.0	0.99220	3.42	0.69	11

In [3]:
```
data.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

Відокремте цільову змінну, розділіть навчальну вибірку у відношенні 7:3 (30% - під задишену вибірку, нехай random_state=17) і нормалізуйте дані за допомогою StandartScaler

In [7]:
```
y = # Ваш код тут
data.drop("quality", axis=1, inplace=True)
X_train, X_holdout, y_train, y_holdout = train_test_split # Ваш код тут
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform # Ваш код тут
X_holdout_scaled = scaler.transform # Ваш код тут
```

Лінійна регресія

Навчіть просту лінійну модель регресії

In [ ]:
```
linreg =   # Ваш код тут
linreg.fit # Ваш код тут
```

> **Питання 1 : Які середньоквадратичні помилки лінійної регресії на навчальній і відкоаденій вибірках ?**

In [ ]:
```
print("Mean squared error (train): %.3f" % # Ваш код тут
print("Mean squared error (test): %.3f" % # Ваш код тут
```

Подивіться на коефіцієнти моделі і ранжуйте ознаки за впливом на якість вина (врахуйте, що великі за модулем негативні значення коефіцієнтів теж говорять про сильний вплив). Створіть для цього новий невеликий DataFrame.

> **Питання 2 : Яку ознаку лінійна регресія вважає найбільш впливовою на якість вина?**

In [ ]:
```
linreg_coef = pd.DataFrame # Ваш код тут
linreg_coef.sort_values # Ваш код тут
```

Lasso-регресія

Навчіть Lasso-регресію з невеликим коефіцієнтом alpha=0,01 (слабка регуляризація). Нехай знову random_state=17.

In [ ]:
```
lasso1 = Lasso # Ваш код тут
lasso1.fit # Ваш код тут
```

Подивіться на коефіцієнти моделі і ранжуйте ознаки за впливом на якість вина. Яка ознака "відпала" першою, тобто найменш важлива для пояснення цільової змінної в моделі Lasso?

In [ ]:
```
lasso1_coef = pd.DataFrame # Ваш код тут
lasso1_coef.sort_values # Ваш код тут
```

Тепер визначте краще значення alpha в процесі 5-кратної крос-валідації. Використовуйте LassoCV і random_state = 17.

In [ ]:
```
alphas = np.logspace(-6, 2, 200)
lasso_cv = LassoCV # Ваш код тут
lasso_cv.fit # Ваш код тут
```

In [ ]:
```
lasso_cv.alpha_
```

Виведіть коефіцієнти "кращого" Lasso в порядку зменшення впливу на якість вина.

> **Питання 3: Яка ознака "занулилася першою" в налаштованій моделі LASSO?**

In [ ]:
```
lasso_cv_coef = pd.DataFrame # Ваш код тут
lasso_cv_coef.sort_values # Ваш код тут
```

Оцініть середньоквадратичну помилку моделі на навчальній і тестовій вибірках.

> **Питання 4 : Які середньоквадратичні помилки налаштованої LASSO-регресії на навчальній і відкладеній вибірках?**

In [ ]:
```
print("Mean squared error (train): %.3f" % # Ваш код тут
print("Mean squared error (test): %.3f" % # Ваш код тут
```

Випадковий ліс

Навчіть випадковий ліс з параметрами "з коробки", фіксуючи тільки random_state=17.

In [ ]:
```
forest = RandomForestRegressor # Ваш код тут
forest.fit # Ваш код тут
```

> **Питання 5 : Які середньоквадратичні помилки випадкового лысу на навчальній вибірці, на крос-валідації (cross_val_score з scoring='neg_mean_squared_error' і іншими параметрами за замовчуванням) і відкладеній вибірках?**

In [ ]:
```
print("Mean squared error (train): %.3f" % # Ваш код тут
print("Mean squared error (cv): %.3f" % # Ваш код тут
print("Mean squared error (test): %.3f" % # Ваш код тут
```

Налаштуйте параметри min_samples_leaf і max_depth за допомогою GridSearchCV і знову перевірте якість моделі на крос-валідації і на відкладеній вибірках.

In [ ]:
```
forest_params = {'max_depth': list(range(10, 25)),
                 'min_samples_leaf': list(range(1, 8)),
                 'max_features': list(range(6, 12))}
locally_best_forest = GridSearchCV # Ваш код тут
locally_best_forest.fit # Ваш код тут
```

In [ ]:
```
locally_best_forest.best_params_, locally_best_forest.best_score_
```

Нажал результати GridSearchCV в повному не відтворювані (можуть відрізнятися на різних платформах навіть при фіксованому random_state). Тому навчіть ліс з параметрами max_depth=19, max_features=7, і min_samples_leaf=1 (краще в моєму випадку).

> **Питання 6 : Які середньоквадратичні помилки налаштованого випадкового лісу на навчальній вибірці, на крос-валідації (cross_val_score з scoring='neg_mean_squared_error') і на відкладеній вибірках?**

In [ ]:
```
print("Mean squared error (cv): %.3f" % # Ваш код тут
print("Mean squared error (test): %.3f" % # Ваш код тут
```

Оцініть важливість ознак за допомогою випадкового лісу.

> **Питання 7 : Яка ознака виявилася найінформативнішою в налаштованій моделі випадкового лісу?**

In [ ]:
```
rf_importance = pd.DataFrame # Ваш код тут
rf_importance.sort_values # Ваш код тут
```

Зробіть висновки про якість моделей і оцінки впливу ознак на якість вина за допомогою цих трьох моделей