*# Import libraries*

**from** sklearn.linear\_model **import** LinearRegression **as** lm

**from** sklearn.metrics **import** mean\_squared\_error

**from** sklearn.model\_selection **import** train\_test\_split, cross\_val\_score

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** pandas **as** pd

**import** seaborn **as** sns

*# PART 1*

*# Q1*

df **=** pd**.**read\_csv("boston\_csv.csv")

df **=** df**.**replace(to\_replace**=**'na', value**=**np**.**nan)**.**dropna()

df

*# Q2*

print(df**.**describe())

correlation\_matrix **=** df**.**corr()**.**round(2)

sns**.**heatmap(data**=**correlation\_matrix, annot**=True**)

plt**.**show()

*# Q3*

X **=** df["LSTAT"]

y **=** df["MEDV"]

X **=** np**.**array(X)**.**reshape(**-**1,1)

y **=** np**.**array(y)**.**reshape(**-**1,1)

print(X**.**shape)

print(y**.**shape)

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.25, random\_state**=**101)

print(X\_train**.**shape)

print(X\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

reg\_1 **=** lm()

reg\_1**.**fit(X\_train, y\_train)

y\_train\_predict\_1 **=** reg\_1**.**predict(X\_train)

mse **=** (mean\_squared\_error(y\_train, y\_train\_predict\_1))

r2 **=** round(reg\_1**.**score(X\_train, y\_train),2)

print("The model performance for training set")

print("--------------------------------------")

print('Mean Squared Error is {}'**.**format(rmse))

print('R^2 score is {}'**.**format(r2))

print("\n")

*# model evaluation for test set*

reg\_1 **=** lm()

reg\_1**.**fit(X\_train, y\_train)

y\_pred\_1 **=** reg\_1**.**predict(X\_test)

mse **=** (mean\_squared\_error(y\_test, y\_pred\_1))

print("The model performance for training set")

print("--------------------------------------")

print("Mean Squared Error: {}"**.**format(mse))

print("\n")

*# Q4*

features **=** ['LSTAT', 'TAX']

X **=** df[features]

y **=** df["MEDV"]

print(X**.**shape)

print(y**.**shape)

X **=** np**.**array(X)**.**reshape(**-**1,2)

y **=** np**.**array(y)**.**reshape(**-**1,1)

print(X**.**shape)

print(y**.**shape)

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.25, random\_state**=**14)

print(X\_train**.**shape)

print(X\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

reg\_1 **=** lm()

reg\_1**.**fit(X\_train, y\_train)

y\_train\_predict\_1 **=** reg\_1**.**predict(X\_train)

mse **=** (mean\_squared\_error(y\_train, y\_train\_predict\_1))

r2 **=** round(reg\_1**.**score(X\_train, y\_train),2)

print("The model performance for training set")

print("--------------------------------------")

print('Mean Squared Error is {}'**.**format(rmse))

print('R^2 score is {}'**.**format(r2))

print("\n")

*# model evaluation for test set*

reg\_1 **=** lm()

reg\_1**.**fit(X\_train, y\_train)

y\_pred\_1 **=** reg\_1**.**predict(X\_test)

mse **=** (mean\_squared\_error(y\_test, y\_pred\_1))

print("The model performance for training set")

print("--------------------------------------")

print("Mean Squared Error: {}"**.**format(mse))

print("\n")

*# PART 2*

*# 1*

wine **=** pd**.**read\_csv("wine\_data.csv")

*# 2*

wine**.**describe()

*# 3*

features **=** ['Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline']

X **=** wine[features]

y **=** wine["Class"]

X **=** np**.**array(X)**.**reshape(**-**1,13)

y **=** np**.**array(y)**.**reshape(**-**1,1)

print(X**.**shape)

print(y**.**shape)

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3, random\_state**=**14)

print(X\_train**.**shape)

print(X\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

*# 4*

**from** sklearn.neighbors **import** KNeighborsClassifier

knn **=** KNeighborsClassifier(n\_neighbors **=** 5)

knn**.**fit(X **=** X\_train, y **=** y\_train)

print("Predictions form the classifier:")

print(knn**.**predict(testset\_data))

print("Target values:")

print(testset\_labels)

*# 머신러닝 모형끼리 비교*

**from** sklearn.svm **import** SVC *# Supprt Vector Classifier*

model **=** SVC(kernel**=**'linear', C**=**1000)

model**.**fit(learnset\_data, learnset\_labels)

print("Predictions form the classifier:")

print(model**.**predict(testset\_data))

print("Target values:")

print(testset\_labels)