Slip 25

25.1

public class Singleton {

// Declare a volatile instance variable to ensure visibility across threads

private static volatile Singleton instance;

// Private constructor to prevent instantiation from outside the class

private Singleton() {

// Initialization code, if needed

}

// Double-checked locking for thread safety

public static Singleton getInstance() {

if (instance == null) {

synchronized (Singleton.class) {

// Check again inside synchronized block to avoid race condition

if (instance == null) {

instance = new Singleton();

}

}

}

return instance;

}

// Other methods, if any

}

25.2

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import cross\_val\_predict

data = pd.read\_csv(r'kc\_house\_data.csv')

data.head(5);

print(data.shape)

f =

['price','bedrooms','bathrooms','sqft\_living','floors','condition','sqft\_abov

e','sqft\_basement','yr\_built','yr\_renovated']

data = data[f]

print(data.shape)

data = data.dropna()

print(data.shape)

data.describe()

X=data[f[1:]]

y=data['price']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size

=0.2,random\_state=42)

print(X\_train.shape)

print(X\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

lr=LinearRegression()

lr.fit(X\_train,y\_train)

print(lr.coef\_)

y\_test\_predict = lr.predict(X\_test)

print(y\_test\_predict.shape)

g = plt.plot((y\_test-y\_test\_predict),marker='o',linestyle="")

plt.show()

25.3

// Import the HTTP module

const http = require('http');

// Configure the HTTP server to respond with "Hello, World!" to all requests

const server = http.createServer((req, res) => {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello, World!\n');

});

// Listen on port 3000 and IP address 127.0.0.1

const PORT = 3000;

const IP = '127.0.0.1';

server.listen(PORT, IP, () => {

console.log(`Server running at http://${IP}:${PORT}/`);

});