

R - Py 컴퓨팅: Homework 2  
Due: 2021-06-23, LMS 제출, 코드는 부록으로 첨부

## PART1. 와인 클래스에 대한 kNN 알고리즘 적용

### 와인 데이터 구성

The data is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators. There are thirteen different measurements taken for different constituents found in the three types of wine.

이 와인의 화학적 분석 결과는 다음과 같이 정리된다.

Attribute Information:

- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline

와인 종류는 class 변수로 구분되어 있으며, 1, 2, 3 을 각 코드값으로 지니고 있다.

이 와인데이터 (wine\_data.csv) 파일에 대해서 다음과 같은 분석을 행하라.

1. Pandas 를 사용해 wine\_data.csv 파일을 wine 데이터프레임을 만든다.

```
$ python part1.py
  Class  Alcohol  Malic acid  Ash  Alcalinity of ash  Magnesium  ...  Nonflavanoid phenols  Proanthocyanins  Color intensity  Hue  OD280/OD315 of diluted wines  Proline
0      1    14.23      1.71  2.43      15.6      127  ...           0.28           2.29           5.64  1.04           3.92    1065
1      1    13.20      1.78  2.14      11.2     100  ...           0.26           1.28           4.38  1.05           3.40    1090
2      1    13.16      2.36  2.67      18.6     101  ...           0.30           2.81           5.68  1.03           3.17    1185
3      1    14.37      1.95  2.50      16.8     113  ...           0.24           2.18           7.80  0.86           3.45    1480
4      1    13.24      2.59  2.87      21.0     118  ...           0.39           1.82           4.32  1.04           2.93     735
..    ..    ..    ..    ..    ..    ..  ..    ..    ..    ..    ..    ..
173   3    13.71      5.65  2.45      20.5      95  ...           0.52           1.06           7.70  0.64           1.74     740
174   3    13.40      3.91  2.48      23.0     102  ...           0.43           1.41           7.30  0.70           1.56     750
175   3    13.27      4.28  2.26      20.0     120  ...           0.43           1.35           10.20  0.59           1.56     835
176   3    13.17      2.59  2.37      20.0     120  ...           0.53           1.46           9.30  0.60           1.62     840
177   3    14.13      4.10  2.74      24.5      96  ...           0.56           1.35           9.20  0.61           1.60     560
[178 rows x 14 columns]
```

2. Wine 데이터에 describe 메소드를 사용하여 요약통계량을 구한다.

```
$ python part1.py
  Class  Alcohol  Malic acid  Ash  Alcalinity of ash  ...  Proanthocyanins  Color intensity  Hue  OD280/OD315 of diluted wines  Proline
count  178.000000  178.000000  178.000000  178.000000  ...  178.000000  178.000000  178.000000  178.000000  178.000000
mean    1.938202    13.000618    2.336348    2.366517    ...    1.590899    5.058090    0.957449    2.611685    746.893258
std     0.775035     0.811827    1.117146    0.274344    ...    0.572359    2.318286    0.228572    0.709990    314.907474
min     1.000000    11.030000    0.740000    1.360000    ...    0.410000    1.280000    0.480000    1.270000    278.000000
25%     1.000000    12.362500    1.602500    2.210000    ...    1.250000    3.220000    0.782500    1.937500    500.500000
50%     2.000000    13.050000    1.865000    2.360000    ...    1.555000    4.690000    0.965000    2.780000    673.500000
75%     3.000000    13.677500    3.082500    2.557500    ...    1.950000    6.200000    1.120000    3.170000    985.000000
max     3.000000    14.830000    5.800000    3.230000    ...    3.500000    13.000000    1.710000    4.000000    1680.000000
[8 rows x 14 columns]
```

3. 주어진 데이터에 첫 변수, class 를 우리가 예측하는 타겟 변수, y, 나머지 모든 변수를 예측 변수 X 로 활용하고, Train/Test 표본을 70% 와 30%가 되도록 데이터를 구성한다. 각각의 명칭은 X\_train/X\_test, y\_train/y\_test 로 칭한다.

4. Scikit-learn 의 KNeighborsClassifier 를 사용하여 70%인 X\_train 과 y\_train 을 바탕으로 모델을 트레이닝 시킨다.
5. 이 Training 된 모델을 바탕으로 Training Set 에서 예측해보고 원래의 라벨을 얼마나 정확하게 예측하는지 percentage 단위로 출력하고, k=5 로 결정하고 다수결 및 유클리디안 거리를 사용한다. .  
**78.2258064516129**
6. 이 Training 된 모델을 바탕으로 Test Set 에서 원래의 라벨을 얼마나 정확하게 예측하는지 percentage 단위로 출력한다.  
**66.66666666666666**
7. 5-6 번을 k=3 일 경우에 대해서도 시행한다.  
**86.29032258064517**  
**64.81481481481481**
8. 3-6 번에서 X 데이터를 Alcohol, Malic acid, Ash, Alcalinity of ash 네 변수만 활용하여 다시 수행한다.  
**89.51612903225806**  
**75.92592592592592**

콜옵션은 ‘살 수 있는 권리’라는 뜻입니다. 좀 유식하게 표현하면 매입선택권이죠. 예를들어 설명해드리겠습니다. 햇빛 내리는 따스한 봄날에 나들이를 하던 농산물 수집상은 걱정거리가 하나 있었습니다. 올해 가을에 배추 값이 폭등할 것 같았기 때문이죠. 한 포기에 최소한 만원은 할 것 같습니다. 이때 농산물수집상은 농부아저씨와 이렇게 계약을 합니다.

“농부아저씨! 올해 가을에 배추 한포기에 2000 원씩, 총 1 만 포기를 사고 싶어요. 하지만 조건이 있어요. 저는 사기 싫으면 안사도 되는 걸로 해요. 제 마음대로 하는 대신에 특별히 보너스 200 만원을 드릴게요”

배추 값이 폭등해 한포기당 1 만원이 되면 어떤일이 일어날까요? 콜옵션, 쉽게말해 ‘살 수 있는 권리’를 행사하면 됩니다. 배추 한포기에 2000 원에 살 수 있는 권리를 행사해 1 만 포기를 사면 됩니다. 물론 배추 값이 폭락하면 특별보너스 200 만원을 깨끗이 포기하면 됩니다.

풋옵션은 ‘팔 수 있는 권리’라는 뜻입니다. 좀 그럴듯하게 표현하면 매도선택권이죠. 예를들어 여러분이 농부 아저씨라고 가정해 봅시다. 봄에 농사를 시작하면서 이만저만 걱정이 아닙니다. 열심히 농사를 지었는데 가을에 배추값이 폭락하면 큰일이기 때문입니다. 그동안의 경험으로 볼 때 올해는 틀림없이 한포기당 10 원까지 같것같은 예감이 듭니다. 이때 농부아저씨는 농산물 수입상과 이렇게 계약을 합니다. “농산물 수집상님! 올해 가을에 배추 한포기에 2000 원, 총 1 만포기를 팔고 싶어요. 하지만 조건이 있어요. 저는 팔기 싫으면 안 팔아도 되는 걸로 해요. 제 마음대로 하는 대신에 특별보너스로 200 만원을 드릴게요”

시간이 흘러 낙엽지는 가을이 찾아왔고, 농부아저씨의 예상대로 배추값이 폭락해 한포기당 10 원이 되었다고 합니다. 어떤일이 일어날까요? 농부아저씨는 풋옵션, 즉 ‘팔 수 있는 권리’를 행사하면 됩니다. 다른 농부아저씨들이 차마 10 원에 배추를 팔 수 없어 밭을 갈아 엮을때! 1 포기에 2000 원씩, 총 1 만포기를 팔아 돈을 벌면 됩니다.

만약 배추가격이 폭등해 한포기당 만원이 되었다면 어떻게 할까요? 그냥 계약을 파기하고, 시장에서 한포기당 만원을 받고 팔면 됩니다. 농부 아저씨에게는 농산물 수집상한테 ‘팔고 싶으면 팔고, 팔기 싫으면 안 팔아도 되는 권리’가 있기 때문입니다.

보다 구체적으로 콜옵션은 기준자산(underlying asset)을 만기일(maturity date/expiration date)에 행사가격(strike price/exercise price)을 주고 살 수 있는 권리라고 정의할 수 있다. 예를 들어 (기준자산, 만기일, 행사가격)이 (A 주식, 7월 12일, 50,000원)으로 정해진 콜옵션을 매입한 사람은 7월 12일에 A 주식을, 그 날의 실제 주가가 얼마이든지 50,000원에 살 수 있는 권리를 가진 것이다. 예를 들어 7월 12일 A 주식의 가격이 65,000원이 된다면, 이 콜옵션을 보유한 투자자는 65,000원의 가치를 갖는 주식을 50,000원에 살 수 있으므로 65,000원과 50,000원의 차액인 15,000원의 수입으로 얻게 된다. 한편, 7월 12일에 A 주식의 가격이 45,000원이 된다면, 45,000원의 가치를 갖는 주식을 50,000원에 살 수 있는 권리는 아무런 가치도 갖지 않게 된다. 왜냐하면 내가 가진 옵션을 행사해서 50,000에 사는 것보다 주식시장에서 사는 것이 훨씬 더 저렴하기 때문이다.

다시 말해서 콜옵션 매입자는 만기 기준자산의 가격이 행사가격보다 크면 그 차액을 수입으로 얻지만, 행사가격보다 작으면 아무런 수입도 얻지 못하게 된다. 콜 매입자가 만기에 기준자산을 행사가격에 사겠다고 요구하는 것을 콜옵션을 행사한다(exercise)고 표현한다. 콜의 매입자는 만기의 기준자산 가격이 행사가격을 초과할 경우만 그 옵션을 행사할 것이다.

옵션은 하나의 권리가기 때문에, 상응하는 의무를 지게 되는 거래의 상대방이 존재해야 한다. 콜옵션을 소유한 사람이 요구할 경우, 만기일에 기준자산을 행사가격에 매입하는 의무를 지겠다고 약속하는 것을 콜옵션을 발행한다고 표현한다. 콜옵션을 발행하려는 사람이, 그러한 의무를 지겠다는 약속에 대해서, 콜옵션을 매입하려는 사람에게 요구하는 대가를 옵션 프리미엄(option premium)이라고 하며 일종의 옵션의 가격으로 해석이 된다. 옵션을 매입하는 사람은 이러한 가격을 지급하고 만기에 기준자산의 가격에 따라서 플러스 또는 0의 수입을 얻는 것이다.<sup>1</sup>

---

<sup>1</sup> 용어를 정리하면, 옵션의 매입자(option buyer/holder)는 옵션에 대해서 사자(long) 포지션을, 옵션의 발행자(option writer/seller)는 팔자(short) 포지션을 취하였다고 표현한다.

수식으로 표현하여 행사가격을  $K$ , 만기일의 기준자산 가격을  $S_T$ 라고 표시할 때, 콜의 소유자는  $\text{Max}(S_T - K, 0)$ 의 수입을 얻게 된다.  $\text{Max}(X, Y)$ 란  $X$ 와  $Y$ 의 두 값 가운데 큰 값을 의미한다. 유사하게  $\text{Min}(X, Y)$ 는 둘 중 작은 값을 의미한다.

정리하자면, 오늘 콜옵션의 가격을  $C$ 로 표현할 때 콜의 소유자와 발행자 각각의 거래일의 현금흐름과 만기에서의 수입은 다음과 같다. 즉

콜옵션 소유자

$$\text{오늘의 현금흐름}^2 = -C$$

$$\text{만기의 수입} = \text{max}(S_T - K, 0)$$

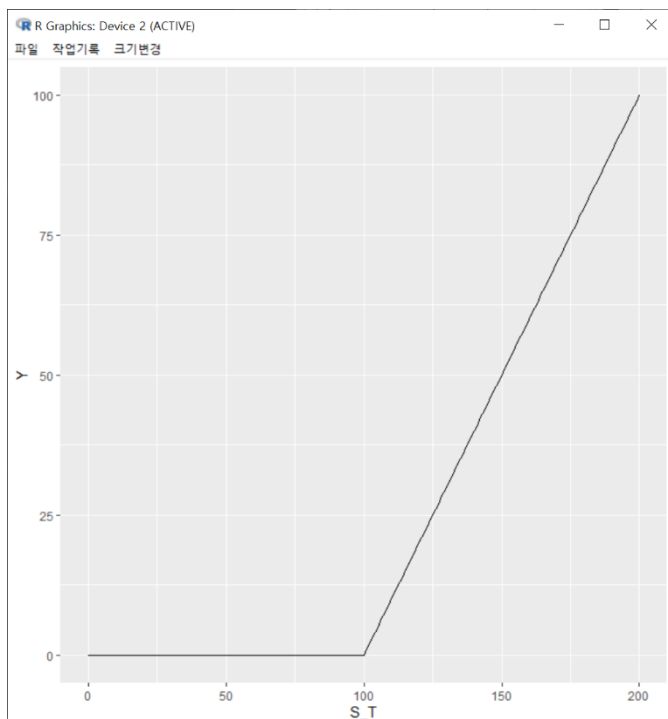
콜옵션 발행자

$$\text{오늘의 현금흐름} = +C$$

$$\text{만기의 수입} = -\text{max}(S_T - K, 0) \text{ 또는 } \text{min}(K - S_T, 0)$$

다음은 R 코드로 작성하고 실행 결과를 보고하라.

**Q1** 만기에서의 콜옵션 소유자의 수입  $Y$ 를 기준자산, 여기서는 주식  $S_T$ 의 가격의 그래프로 나타내라.  $S_T$ 의 범위는 0 부터 200 까지이며  $K=100$ 으로 고정한다



<sup>2</sup> 오늘 옵션 거래가 일어났다고 생각한다.

**Q2.** 노벨상을 수상한 블랙-숄즈-머턴은 이와 같은 콜옵션의 프리미엄 C 가 특정가정들을 만족한 경우 다음과 같은 식에 의해서 결정되어야 한다고 증명하였다.

### ● B-S 옵션가격결정모형

$$C = S \cdot N(d_1) - Ke^{-r_f T} \cdot N(d_2)$$

**C**: 콜옵션의 가격      **S**: 주식의 가격

**K**: 행사가격      **r<sub>f</sub>**: 무위험이자율

**T**: 옵션의 만기      **σ**: 주식수익률의 연간 표준편차

**N(d)**: 표준정규분포의 z=d까지의 누적확률

$$d_1 = \frac{\ln(S/K) + \left(r_f + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \quad d_2 = d_1 - \sigma\sqrt{T}$$

여기서 S 는 오늘의 주식 가격을 의미하고 e 는 자연대수를 의미한다. 따라서 오늘의 콜옵션가격은 오늘의 주가(S), 만기(T), 행사가격(K), 주식의 변동성 σ, 무위험 이자율 r<sub>f</sub>에 의해 결정된다. 이 다섯가지 요소를 함수의 인자로 받고, 콜옵션 가격을 계산해주는 BSM 함수를 R 로 코딩한 후, S=100, K=100, T=0.5, σ=0.1, r<sub>f</sub>=0.04 에 대해 결과값을 계산하라.

**D1 = 0.3181981**

**D2 = 0.2474874**

**C = 3.893411**

**Q3.** 재무 이론의 발전은 블랙숄즈머턴의 방법론과 동일한 결과를 다음과 같이 옵션 만기

T 시점까지 주가의 경로를 N 번 시뮬레이션을 함으로써 얻을 수 있음을 확인하였다. 여기서

i 는 i 번째 주가의 path 를 의미하고 t 는 0 부터 만기 T 까지의 시간 경로를 의미한다. 주가의

경로를 0, 0+Δt, 0+2Δt,..... 0+MΔt, Δt=T/M 시점에 대해서 각각 산출한다는 의미이다. 이와

같은 시점별 주가 S<sub>t</sub><sup>i</sup> 즉 t 시점 주가에서 S<sub>t+Δt</sub><sup>i</sup>로 t+Δt 시점으로의 주가 경로식은 아래와 같이

주어진다.

$$S_{t+\Delta t}^i = S_t^i \exp \left[ (r - 0.5\sigma^2)\Delta t + \sigma\sqrt{\Delta t}\epsilon_t^i \right]$$

$$\epsilon_t^i \sim N(0,1)$$

이러한  $i$  번째 주가 경로에 대해서 만기시점의 옵션의 가치  $V_i$  는

$$V^i = \max(S_T^i - K, 0)$$

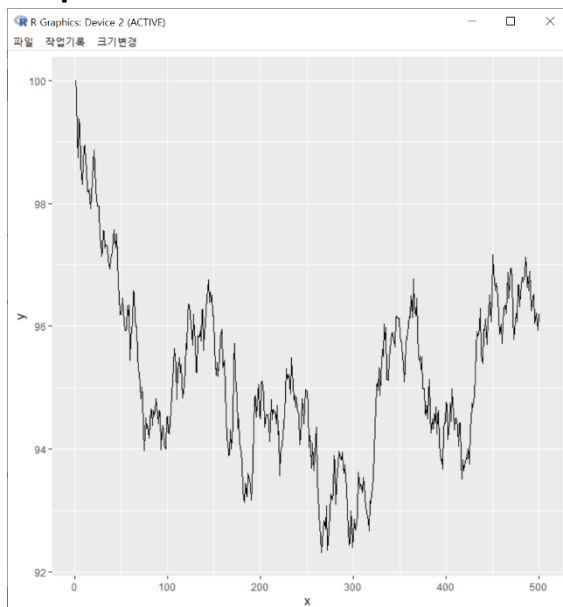
과 같이 정리될수 있다. 최종적으로 옵션의 가치  $C$  는

$$C = \exp(-r_f T) \frac{1}{N} \sum_{i=1}^N V^i$$

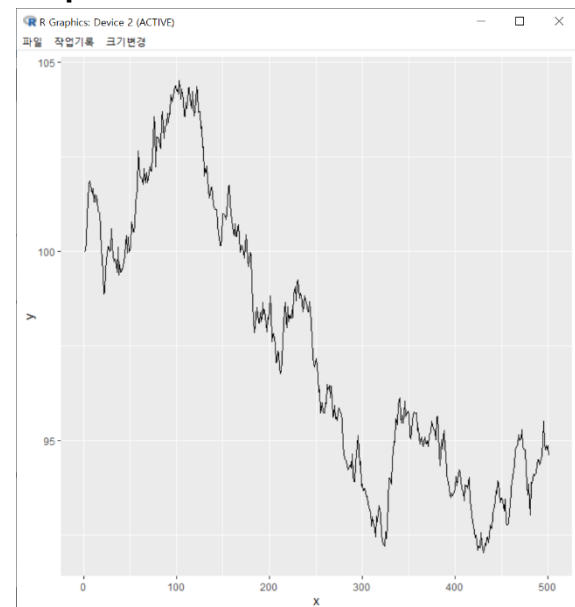
로 계산된다.  $\Delta t = 1/1000$ ,  $N = 10000$  번으로 하여 Q2 와 동일한 상황에서 옵션 가격을 계산한다 즉 (500+1)개의 관측치를 가진 주가의 예상 경로를 10000 번 생성하여 T 기에서 주식가격 10000 개 생성하고 그에 따르는 옵션가격을 결정한다. 또한 10000 개중 다섯개의 주가 경로에 대해 오늘부터 만기까지(0->T) 꺾은선 형태의 그래프로 표기한다. 여기서 X 축은 0,  $0 + \Delta t$ ,  $0 + 2\Delta t$ ,.....  $0 + M\Delta t$ ,  $\Delta t = T/M$  이고 대응되는 Y 값은 각 시점의 경로별 주가이다.

**C : 3.824025**

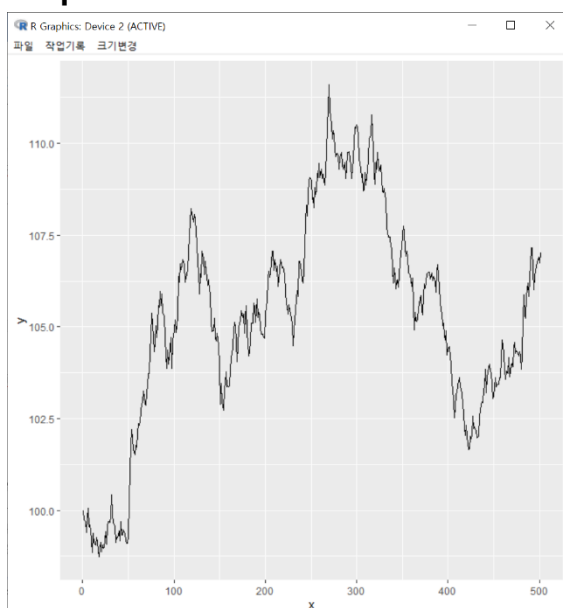
**Graph1**



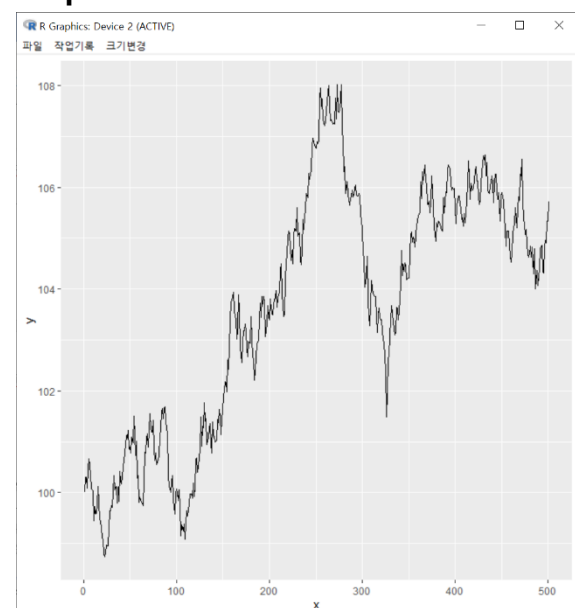
**Graph2**



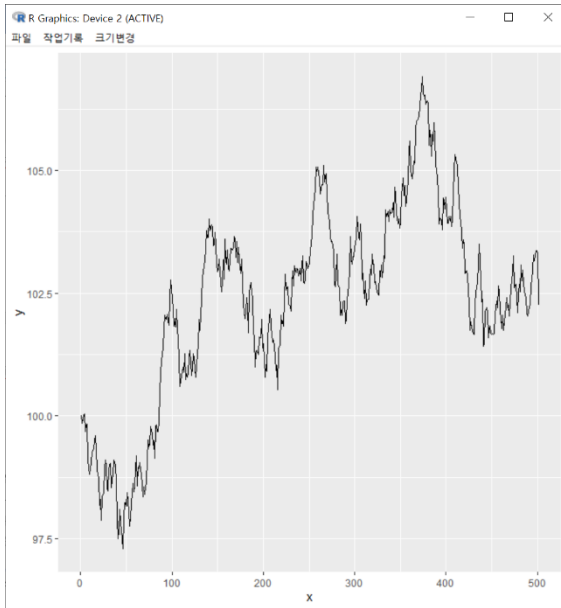
**Graph3**



**Graph4**



## Graph5



## 부록

### PART1

```
# part1
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pandas as pd

#1
wine = pd.read_csv('C:/Users/user/Desktop/대학교/21-1 학기/AI+X/중간과제 2/part1/wine_data.csv')
# print(wine)

#2
wine.describe()
# print(wine.describe())

#3
lm = LinearRegression()
X = wine[wine.columns.difference(["Class"])]
Y = wine["Class"]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=19)

#4
knn = KNeighborsClassifier()

#5
knn = KNeighborsClassifier(n_neighbors=5, p=2)
knn.fit(X_train, y_train)
y_pred_train = knn.predict(X_train)
```



```

print(((np.mean(y_pred_train == y_train))*100))

#6
y_pred_test = knn.predict(X_test)
print(((np.mean(y_pred_test == y_test))*100))

#7
knn = KNeighborsClassifier(n_neighbors=3, p=2)
knn.fit(X_train, y_train)
y_pred_train = knn.predict(X_train)
print(((np.mean(y_pred_train == y_train))*100))
y_pred_test = knn.predict(X_test)
print(((np.mean(y_pred_test == y_test))*100))

#8
lm = LinearRegression()
X = wine[["Alcohol", "Malic acid", "Ash", "Alcalinity of ash"]]
Y = wine["Class"]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=19)
knn = KNeighborsClassifier(n_neighbors=5, p=2)
knn.fit(X_train, y_train)
y_pred_train = knn.predict(X_train)
print(((np.mean(y_pred_train == y_train))*100))
y_pred_test = knn.predict(X_test)
print(((np.mean(y_pred_test == y_test))*100))

```

## PART2

### Q1

```

# Q1
library("ggplot2")
K = 100
S_T <- c(0:200)
Y <- c()
for (i in S_T + 1) {
  Y[i] <- max(S_T[i] - K, 0)
}
df <- data.frame(S_T, Y)
ggplot(data = df, mapping = aes(x =S_T, y=Y)) + geom_line()

```

### Q2

```

# Q2
BSM <- function(S,K,T,sigma,rf) {
  d1 <- cal_d1(S,K,T,sigma,rf)
  d2 <- cal_d2(d1,T,sigma)

  C = S * pnorm(d1) - K * exp(-1*rf*T) * pnorm(d2)
  return(C)
}

```

```

cal_d1 <- function(S,K,T,sigma,rf){
  d1 <- (log(S/K) + (rf+(1/2)*sigma^2)*T) / (sigma*sqrt(T))
}

cal_d2 <- function(d1,T,sigma) {
  d2 <- d1 - sigma*sqrt(T)
  return(d2)
}

print(BSM(100, 100, 0.5, 0.1, 0.04))

```

### Q3

```

# Q3
library("ggplot2")

# 정해진 것들 세팅 : M=500, T=0.5, dt = 0.5/500, N=100, rf=0.04
S0 = 100
K = 100
T = 0.5
M = 500
sigma = 0.1
rf = 0.04
N = 10000
dt = T/M

# 한컬럼이 하나의 시계열 경로
# 행의 숫자는 시계열의 수
# 칼럼의 숫자는 시뮬레이션 횟수

sim_S = matrix(0, nrow=501, ncol=N)

sim_S[1,] = S0

for (nn in 1:N) {
  for (t in 1:M) {
    sim_S[t+1,nn] = sim_S[t,nn]*exp((rf-0.5*sigma^2)*dt+sigma*sqrt(dt)*rnorm(1))
  }
}

S_T = sim_S[M+1,]
V = rep(0,N)

for (i in 1:N) {
  V[i] = max((S_T[i]-K), 0)
}

C = exp(-1*rf*T) * (1/N) * sum(V)
print(C)

```

```
f <- function(i, S) {  
  x <- 1:nrow(S)  
  y <- S[,i]  
  
  df = data.frame(x,y)  
  ggplot(data = df,mapping = aes(x=x,y=y)) + geom_line()  
}  
  
f(1,sim_S)  
f(2,sim_S)  
f(3,sim_S)  
f(4,sim_S)  
f(5,sim_S)
```