Amogha Sekhar, A53301791, CSE 250

Question 4.4

Part A

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
with open('nasdaq00.txt', 'r') as f:
    data_2000 = f.read().split('\n')
```

In [3]:

```
for i in range(len(data_2000)):
    data_2000[i]= float(data_2000[i])
#print(data_2000)
```

In [4]:

```
with open('nasdaq01.txt', 'r') as f:
    data_2001 = f.read().split('\n')
```

In [5]:

```
for i in range(len(data_2001)):
    data_2001[i]= float(data_2001[i])
#print(data_2001)
```

In [6]:

```
data_00= np.array(data_2000)
#print(data_00)
```

In [7]:

```
data_01= np.array(data_2001)
#print(data_01)
```

In [8]:

```
X_00= []
Y_00= []
for i in range(len(data_00)-3):
    X_00.append([data_00[i], data_00[i+1], data_00[i+2]])
    Y_00.append(data_00[i+3])

#print(X_00)
#print(Y_00)

X= np.array(X_00)
Y= np.array(Y_00)
```

In [9]:

```
A= np.matmul(X.transpose(), X)
print(A.shape)
print(A)
```

```
(3, 3)
[[  3.65159733e+09   3.64553007e+09   3.64024608e+09]
 [  3.64553007e+09   3.64300188e+09   3.63752268e+09]
 [  3.64024608e+09   3.63752268e+09   3.63556951e+09]]
```

In [10]:

```
B= np.matmul(Y, X)
```

In [11]:

```
print(B)
```

```
[  3.63403032e+09   3.63120829e+09   3.62909744e+09]
```

In [12]:

```
A_inverse= np.linalg.inv(A)
```

In [13]:

```
w_ML= np.matmul(A_inverse, B)
print(w_ML)
print(w_ML.shape)
```

```
[ 0.03189569  0.01560133  0.95067337]
(3,)
```

Answer for Part A

The linear coefficients are a1= 0.95067337, a2= 0.01560133, a3= 0.03189569

Part B

In [14]:

```
pred= []
for i in range(len(X)):
    val= np.matmul(X[i], w_ML)
    pred.append(val)
```

In [15]:

```
#print(pred)
```

In [16]:

```
#print(Y)
```

In [17]:

```
MSE_00= sum((pred-Y)**2)
n= len(pred)
MSE_2000= MSE_00/n
```

In [18]:

```
print(MSE_2000)
```

```
13902.4010764
```

In [19]:

```
X_01= []
Y_01= []
for i in range(len(data_01)-3):
    X_01.append([data_01[i], data_01[i+1], data_01[i+2]])
    Y_01.append(data_01[i+3])


X_1= np.array(X_01)
Y_1= np.array(Y_01)
```

In [20]:

```
pred_01= []
for i in range(len(X_1)):
    val= np.matmul(X_1[i], w_ML)
    pred_01.append(val)
```

In [21]:

```
#print(pred_01)
```

In [22]:

```
MSE_01= sum((pred_01-Y_1)**2)
n= len(pred_01)
MSE_2001= MSE_01/n
print(MSE_2001)
```

```
2985.09792411
```

Answer for Part B

The Mean Squared Error for the year 2000: 13902.4010764

The Mean Squared Error for the year 2001: 2985.09792411

I would not recommend this linear model for stock market prediction. This is because, the MSE on the training set is very high. This is due to a lot of fluctuation in the stock market during the year 2000. For the year 2001, the linear model worked better because there was less fluctuation in the data. So, if we were to make a stock market prediction for a year where there is a lot of fluctuation in the data, this linear model would not be helpful.

In [ ]:

```

```