Homework -3 Multiple Linear Regression

Name: Shadeeb Hossain

ID: sh7492

- 1. (a) The target variable could be "Quantity of Products sold".
 - (b) The suggested Linear Model could be equivalent to:

$$y = \beta_1 + \beta_2 x_2 - \beta_3 x_2 - \beta_4 x_2$$

Where, β_1 is the Quantity of Products sold in 1 month.

 β_2 is the frequency of the occurrence of the word "good".

 β_3 is the frequency of the occurrence of the word "bad".

 β_4 is the frequency of the occurrence of the word "does not work".

 x_2 is the number of total reviews.

- (c) Normalization can be used as shown below:
- user_input=int(input("Please enter the total score (out of 5 or 10):"))
 print("The total score is out of :",(user_input))
- Please enter the total score (out of 5 or 10):5 The total score is out of : 5

```
if user_input==5:
    actual_score=int(input("Enter how much the product scored out of 5:"))
    score=actual_score/5
    print("The normalized score is:",score)
else:
    actual_score=int(input("Enter how much the product scored out of 10:"))
    score=(actual_score/10)*2
    print("The normalized score is:",score)
```

Enter how much the product scored out of 5:4 The normalized score is: 0.8

(d) For missing data. dropna() command can be used as shown below

```
# (d)
df1=df[['review']]
df2=df1.dropna()
```

(e) Fraction of reviews with the word "good" is a better choice between the two. This is because it accounts for normalization. Some products might have a higher number of "good" reviews, but their "bad" reviews might also be significantly higher. This error can be reduced by opting for fraction of reviews with the word "good".

2. (a) Linear model for y in terms of x_1 and x_2 is:

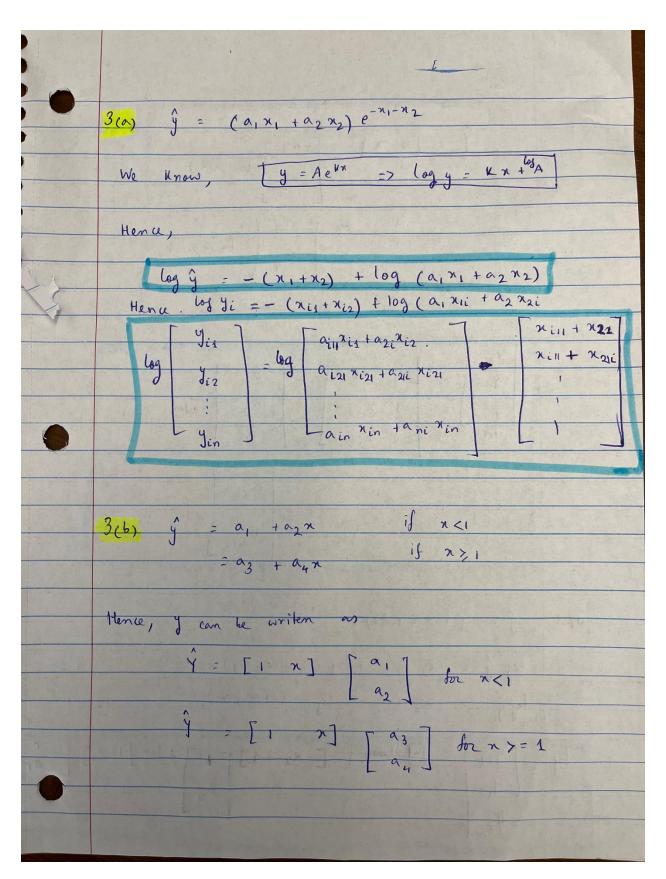
$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}$$

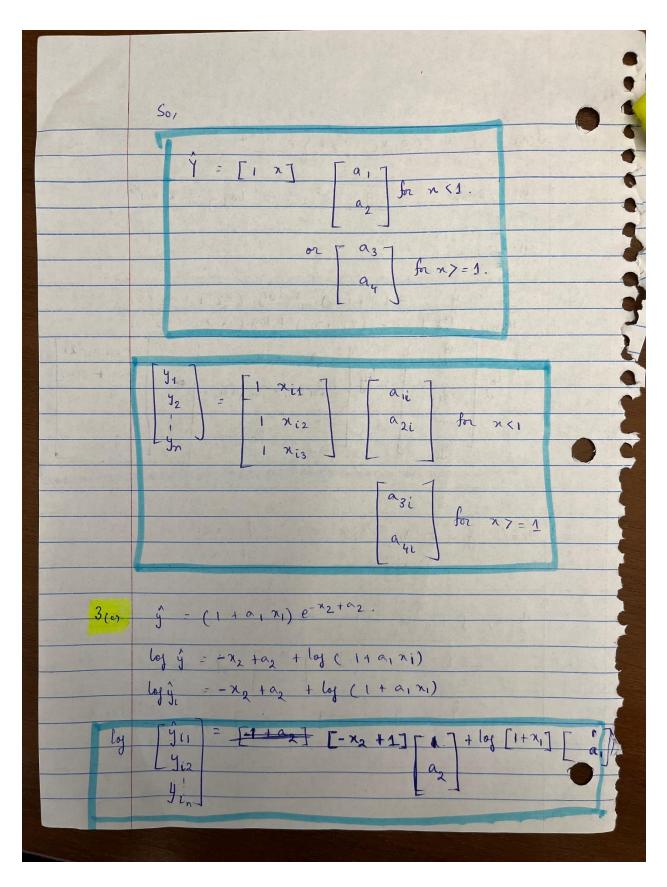
(b)

```
b=np.array([[0,0,1],
           [0,1,4],
           [1,0,3],
           [1,1,7]])
# Save array into a CSV file
np.savetxt("data1.csv",b)
names=['x1','x2','y']
df=pd.read_csv('data1.csv',header=None,delim_whitespace=True,names=names,na_values='?')
df.head()
df2=df['y']
df1=df[['x1','x2']]
print(df1)
ym=np.mean(df2)
y1=df2-ym
xm=np.mean(df1,axis=0)
x1=df1-xm[None,:]
# Computing the correlation
syy=np.mean(y1**2)
sxx=np.mean(x1**2,axis=0)
sxy=np.mean(x1*y1[:,None],axis=0)
# Computing the coefficients
beta1=sxy/sxx
beta0=ym-beta1*xm
Rsq=sxy**2/sxx/syy
print("The value of beta0 is", beta0)
print("The value of beta1 is",beta1)
print("The value of Rsq is",Rsq)
```

```
x2
   x1
0 0.0 0.0
1 0.0 1.0
2 1.0 0.0
  1.0 1.0
The value of beta0 is x1
                           2.5
x2
     2.0
dtype: float64
The value of beta1 is x1
                           2.5
x2
     3.5
dtype: float64
The value of Rsq is x1 	 0.333333
x2
     0.653333
```

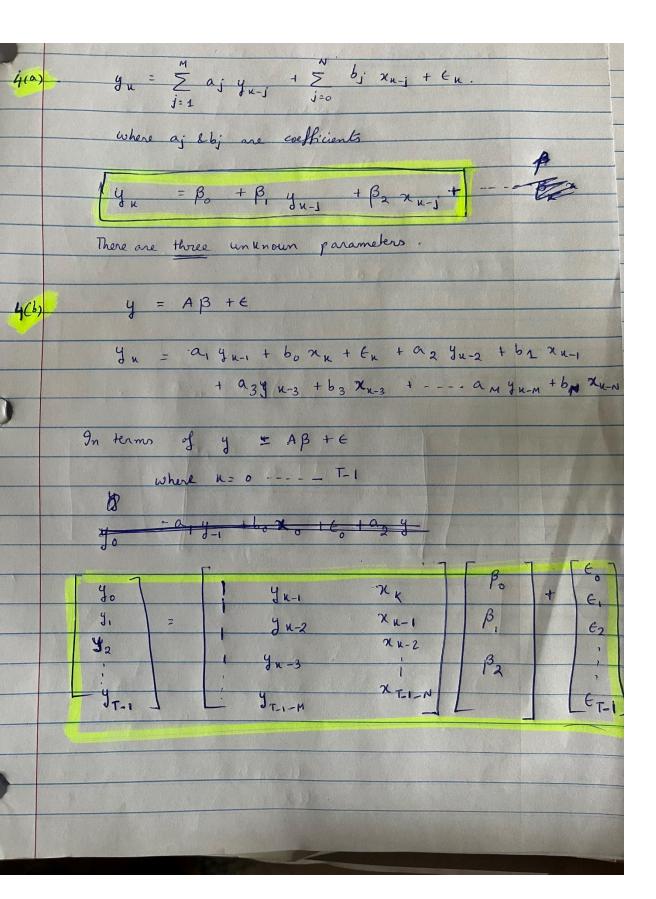
3.





Please turn to next page

Page left intentionally blank



Please continue to the next page for Number 5

Blank Page

1 x grand work & 1	petalighanil and ho	
2 million or a water	the ist to but the ma	
5.(a) (i) n = \$	[a]	
e=1	Cos (rek) + be Sin (Rek)	
Where I = no of homes present		
Re-toral frequencies		
are & be are coefficients		
1 2n where	Was a second	
Γ 2 n where κ=0 N-1.		
x = a = (((((((((((((((((
0 (0 (0) + 60 510 (50 0) +		
6 6	21.1) + b1 Sin (11.1) +	
	$n_2.2) + b_2 Sin (n_2.2) +$	
an les	(1, N) + bN Sin (2N - N).	
	7 [50]	
No Tao	2 + b, Sin Si	
2 = a2 Cos	$2 + b_2 \sin(2)$ $3 + b_2 \sin(2)$	
L nn - Lan Cos	N + bN Sin(N) L SIN	
N. Company of the com	B	
X =		
The 1st sq method can	be used to solve x = AB.	
If returns the least square solution to a linear		
matrix equation.		
N = 100		
ones = np . ones $((N, 1))$		
A = np. hstack ((ones ohm)) # where ohm is equivalent tox		
A. Shape.		

	· · · · · · · · · · · · · · · · · · ·
	out = np. linalg. 1stsq (A, x, reond = None) # beta = out [o] where x is equivalent to y.
	beta.
	The value of B (below) will give the coefficients
	allbe for this model (Ans).
(h)	No, if she is not known, it cannot be
	a linear regression problem. This is
	because for linear regression problem the
1	x train needs to be known to apply. regr = linear model. brearkegression()
(4.	131 162 nd + (n+ 152) 000 110
	05t 0X 0X
	1 0 00 00 00 00 00 00 00 00 00 00 00 00
	CHICA I LOS I LINE CON I SENTE
	A STATE OF THE STA
	The 61 sq metal in be will to when some
	I should be been square soldiers to I was
	media de deservir
	(be who enter our of b)
You had a	

6

```
#(a)
import numpy as np
n=X.shape [0]
yhat=np.zeros(n)
# Without 'For' loop and using Vectorization and Python Broadcasting
yhat=np.sum((beta[0]*X[:,0])+(beta[1]*X[:,1])+(beta[2]*X[:,1]*X[:,2]),axis=1) # axis=1 since it is along the rows
```

```
#(b)
n=len(X)
m=len(alpha)
yhat=np.zeros(n)
yhat+=np.sum(alpha*np.exp(-beta*X),axis=1)
```

```
#(c)
n,d=X.shape
m,d=y.shape
dist=np.zeros((n,m))
dist=np.sum((X[:,None]-y[None,:,:])**2,axis=2)
```