BUAN 6383.002 – Modeling for Business Analytics - Project 1- *Group 1*

Siddhesh Madhukar Koli
Chia-Yi Yen
Sonia Carolina Jaime Chinchilla
Sai Kruthik Reddy Paduru
Lakshmi Sandeep Reddy Dasari

Part I: Replicating Models from Class

Question 1:

Consider the hard candy example from class. The associated data is in the file candy.csv. Develop the following models discussed in class using maximum likelihood estimation (MLE) - Report your code and all relevant details, including the estimated values of the parameters for each model and the corresponding log-likelihood values. Please add comments to your code to make it easy to understand:

a) The Poisson Model

You may find the code for the Poisson Model below:

```
#1. we name the arrays that will go into the function
a = np.array(df[['People']])
b = np.array(df[['Packs']])
lmbda = 1

#2.we define the Poisson Regression Model
def PLL(lmbda,a,b):
    #variable c will store the cumulative log-likelihood
    c = 0
    #the for loop below will iterate throuh each row of the dataset
    for i in range(len(a)):
        c += a[i]*np.log(poisson.pmf(b[i], lmbda))
    return (-1)*c

#3.we use the minimize function to optimize the parameter
#(in this case, LAMBDA) to maximize the log likelihood
solpm1 = minimize(
    PLL,
    args = (a,b),
    x0 = np.array((1)),
    bounds=[(0.000001,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)

#4. we print the parameters and final log likelihood value
final_lmbda = solpm1.x[0]
ll_pm1 = solpm1.fun[0]
print("Lambda:{}".format(final_lmbda))
print("Log likelihood:{}".format(ll_pm1*(-1)))
```

```
Lambda:3.9912283512077944
Log likelihood:-1544.9963904489719
```

The corresponding parameters and log-likelihood are as follows:

Model	Poisson
Log-likelihood	-1545
Lambda	3.99

b) The NBD Model

You may find the code for the NBD Model below:

```
#1. define the NBD model function
def NBD(params,k,t):
    alpha,n = params
    if k==0:
        return (alpha/(alpha+t))**n
    else:
        return (((n+k-1)*t)/(k*(alpha+t)))*NBD(params,k-1,t)

#2. define the model that will calculate the cumulative log likelihood of the NBD model
def NBDLL(params,t,a,b):
    nbd = 0
    for i in range(len(a)):
        nbd += a[i]*np.log(NBD(params,b[i],t))
    return (-1)*nbd

#3. run the NBDLL model through the minimize function
#this will optimize the parameters alpha and n in order to maximize the log likelihood
alpha = 1
n = 1
t = 1
solnbd1 = minimize(
    NBDLL,
    args = (t,a,b),
    x0 = np.array((1,1)),
    bounds=[(0.000001,None),(0.000001,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
alpha = solnbd1.x[0]
n = solnbd1.x[1]
ll_nbd1 = solnbd1.fun[0]
print("Alpha:{}".format(alpha))
print("n:{}".format(n))
print("Log likelihood:{}".format(ll_nbd1*(-1)))
```

The corresponding parameters and log-likelihood are as follows:

Model	NBD
Log likelihood	-1140.0237462459445
alpha	0.2499605034092882
n	0.9976364542507823

c) The Zero Inflated NBD Model

You may find the code for the Zero Inflated NBD Model below:

```
candy=pd.read_csv('candy.csv')
k = np.array(candy[['Packs']]) #define the first array that will go through the PLL function
b = np.array(candy[['People']]) #define the second array that will go through PLL

def NBD(params,k):#this function will calculate NBD
    alpha,n,pi = params #three parameters
    if k==0:
        return ((alpha/(alpha+1))**n)
    else:
        return (((n+k-1)/(k*(alpha+1)))*NBD(params,k-1))

def ZINBD(params,k):#this function will calculate the ZERO INFLATED NBD by utilizing the NBD function
    alpha,n,pi= params
    if k==0:
        return pi +(1-pi)*NBD(params,k)
    else:
        return (1-pi)*NBD(params,k)

def ZINBDLL(params,k,b): #this function will calculate the log likelihood of a ZI NBD by utilizing the ZINBD function
    alpha,n,pi=params
    zinbd = 0
    for i in range(len(b)):
        zinbd += b[i]*np.log(ZINBD(params,k[i]))
    return (-1)*zinbd

alpha = 1 # 000001 -inf
n = 1 # 000001 -inf
pi =0.5 # 000001 -0.99999
solzinbd = minimize(
    ZINBDLL,
    args = (k,b),
    x0 = np.array((1,1.5,0.00000001)),
    bounds=[(0.000001,None),(0.000001,None),(0.000001,0.999999)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	Zero Inflated NBD model
Log likelihood	-1136.16564083
alpha	0.33418767
n	1.50392276
pi	0.11310436

d) The Finite Mixture Models for 2, 3, and 4 segments

a. 2-segment

You may find the code for the Finite Mixture Model, with 2 segments, below:

```
def two_segment(params,a,b):
    pi,lambda1,lambda2 = params
    ll = 0
    for i in range(len(b)):
        ll += a[i]*np.log(pi*(poisson.pmf(b[i],lambda1)) + (1-pi)*(poisson.pmf(b[i],lambda2)))
    return (-1)*ll

a = np.array(df[['People']])
b = np.array(df[['Packs']])
sol2segment = minimize(
    two_segment,
    args = (a,b),
    x0 = np.array((1,1,1)),
    bounds=[(0.000001,0.999999),(0.000001,None),(0.000001,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	Finite Mixture model, 2-segment
Log likelihood	-1188.83
Lambda 1	1.802
Lambda 2	9.121
pi	0.701

b. 3-segment

You may find the code for the Finite Mixture Model, with 3 segments, below:

```
def three_segment(params,a,b):
    theta1,theta2,lambda1,lambda2,lambda3 = params
    pi1 = (e**theta1)/((e**theta1)+(e**theta2)+1) # calculating pie values(pie1,pie2,pie3)
    pi2 = (e**theta2)/((e**theta1)+(e**theta2)+1)
    pi3 = (1)/((e**theta1)+(e**theta2)+1)
    ll = 0
    for i in range(len(b)):
        ll += a[i]*np.log(pi1*(poisson.pmf(b[i],lambda1)) + pi2*(poisson.pmf(b[i],lambda2)) + pi3*(poisson.pmf(b[i],lam
    return (-1)*ll


a = np.array(df[['People']])
b = np.array(df[['Packs']])
sol3segment = minimize(
    three_segment,
    args = (a,b),
    x0 = np.array((1,2,1,1,1)),
    bounds=[(None,None),(None,None),(0.000001,None),(0.000001,None),(0.000001,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	Finite Mixture model, 3-segment
Log likelihood	-1132.04
Lambda 1	0.291
Lambda 2	3.483
Lambda 3	11.216
Theta 1	0.430
Theta 2	1.105

c. 4-segment

You may find the code for the Finite Mixture Model, with 4 segments, below:

```
def four_segment(params,a,b):
    theta1,theta2,theta3,lambda1,lambda2,lambda3,lambda4 = params
    pie1 = (e**theta1)/((e**theta1)+(e**theta2)+(e**theta3)+1) # calculating pie values(pie1,pie2,pie3)
    pie2 = (e**theta2)/((e**theta1)+(e**theta2)+(e**theta3)+1)
    pie3 = (e**theta3)/((e**theta1)+(e**theta2)+(e**theta3)+1)
    pie4 = (1)/((e**theta1)+(e**theta2)+(e**theta3)+1)
    ll = 0
    for i in range(len(b)):
        ll += a[i]*np.log(pie1*(poisson.pmf(b[i],lambda1)) + pie2*(poisson.pmf(b[i],lambda2)) + pie3*(poisson.pmf(b[i],
    return (-1)*ll

a = np.array(df[['People']])
b = np.array(df[['Packs']])
sol4segment = minimize(
    four_segment,
    args = (a,b),
    x0 = np.array((1,2,3,1,1,1,1)),
    bounds=[(None,None),(None,None),(None,None),(0.000001,None),(0.000001,None),(0.000001,None),(0.000001,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	Finite Mixture model, 4-segment
Log likelihood	-1130.07
Lambda 1	3.002
Lambda 2	0.205
Lambda 3	7.418
Lambda 4	12.872
Theta 1	1.598
Theta 2	0.876
Theta 3	0.398

Question 2:

Evaluate the models developed; explain which of them is best, and why. Are there any significant differences among the results from these models? If so, what exactly are these differences? Discuss what you believe could be causing the differences.

Which is the best model and why?

In the table below, we have arranged the results of the 6 models developed so far, and evaluated each with AIC, BIC and X-square p-value:

Model	LL	# of parameters	AIC	BIC	X square p_value
Poisson	-1545	1	3091	3096	0.00
NBD	-1140	2	2284	2292	0.132
ZI-NBD	-1336	3	2278	2290	0.231
2-segment	-1188	3	2383	2396	0.00
3-segment	-1132	5	2274	2294	0.458
4-segment	-1130	7	2274.1	2302	0.709

Based on different criteria, the best models are as follows:

Criteria	Best choice	Second best choice
Log-likelihood ratio	4 segment FM	3 segment FM
AIC	3 segment FM	4 segment FM
BIC	ZI-NBD	NBD, 3 segment FM

Based on the results of our models and our evaluations, we recommend choosing the 3-segment Finite Mixture Model as the best model because it offers the lowest AIC value, the second highest log-likelihood ratio and the third lowest BIC value among all the other models. It is worth mentioning that the 4-segment Finite Mixture Model offers the highest log-likelihood ratio, however, we don't view it as the best model because the number of parameters it uses are way too many than any other models. Furthermore, the p value (0.709) is too high to reject the null hypothesis.

Any significant differences among the results from these models? What exactly are these differences?

Difference A: First, we noticed that the log-likelihood significantly improved from the Poisson model to the NBD model:

Model	LL
Poisson	-1544
NBD	-1140

By observing the log-likelihood ratio, we can see that the NBD model outperforms the Poisson model. We believe that's because the parameters n and alpha from the NBD model help us further capture heterogeneity from the original records; therefore, the model can better fit the original datasets.

Difference B: Second, we noticed that the more segments we include in a Finite Mixture model, the higher the log-likelihood ratio:

Model	LL
2-segment FM	-1188
3-segment FM	-1132
4-segment FM	-1130

We believe that the more segments we use, the finer classification is applied to original records; therefore, the model can be more consistent with the original data. We must note that this method is prone to overfitting, so we must weigh the effectiveness of the model with its performance on the classification of new data.

Question 3:

Based on the 2, 3, and 4-segment finite mixture models, how many packs are the following customers likely to purchase over the next 8 weeks?

a) a customer who purchased 5 packs in the past week, and

Model	Prediction for someone who purchased 5 packs	Prediction for someone who purchased 9 packs, if they're in segment 1
FM 2-segment	43	14
FM 3-segment	31	28
FM 4-segment	34	24

b) a customer who purchased 9 packs in the past week.

Model	Prediction for someone who purchased 9 packs	Prediction for someone who purchased 9 packs, if they're in segment 1
FM 2-segment	73	73
FM 3-segment	80	90
FM 4-segment	69	59

You may find the code used for this question below:

For question 3, a)

2-segment prediction

```
#2 segments
lambda2 = 1.802152486641123
lambda1 = 9.120665766225097
pie1 = 0.2991217614346511
pie2 = 0.7008782385653489

prob_s1_5 = (pie1*poisson.pmf(5,lambda1))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2))
prob_s2_5 = (pie2*poisson.pmf(5,lambda2))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2))
print("Expected purchases next 8 weeks if past week = 5 packs is {}".format(8*(lambda1*(prob_s1_5) + lambda2*(prob_s2_5
maxi = max(prob_s1_5,prob_s2_5)
print("Expected purchases next 8 weeks if past week = 5 packs, if customer is classified into segment2(since p(s=1,x=5)
```

```
Expected purchases next 8 weeks if past week = 5 packs is 42.781266102377096
Expected purchases next 8 weeks if past week = 5 packs, if customer is classified into segment2(since p(s=1,x=5) < p
(s=2,x=5)) is 14.417219893128983
```

3-segment prediction

```
#3segments
lambda1 = 3.483321426064036
lambda2 = 11.2158213224157
lambda3 = 0.29055258688349966
theta1 = 0.6744334195584446
theta2 = -0.43040650590390545
theta3 = 0.0
pie1 = e**(theta1) / ((e**theta1) + (e**theta2) + (e**theta3))
pie2 = e**(theta2) / ((e**theta1) + (e**theta2) + (e**theta3))
pie3 = e**(theta3) / ((e**theta1) + (e**theta2) + (e**theta3))

prob_s1_5 = (pie1*poisson.pmf(5,lambda1))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
prob_s2_5 = (pie2*poisson.pmf(5,lambda2))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
prob_s3_5 = (pie3*poisson.pmf(5,lambda3))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
print("Expected purchases next 8 weeks if past week = 5 packs is {}".format(8*(lambda1*(prob_s1_5) + lambda2*(prob_s2_5
maxi = max(prob_s1_5,prob_s2_5,prob_s3_5)
print("Expected purchases next 8 weeks if past week = 5 packs, if customer is classified into segment1(since prob(s=2,x
```

```
Expected purchases next 8 weeks if past week = 5 packs is 30.825233285033676
Expected purchases next 8 weeks if past week = 5 packs, if customer is classified into segment1(since prob(s=2,x=5) &
prob(s=3,x=5) < prob(s=1,x=5)) is 27.86657140851229
```

```
#4segment
lambda1 = 7.417654014502218
lambda2 = 0.2047757185785757
lambda3 = 12.872409491251577
lambda4 = 3.0019275641803396
theta1 = -1.2001167375208144
theta2 = -0.7220301738582039
theta3 = -1.5980245259938597
theta4 = 0.0
pie1 = e**(theta1) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie2 = e**(theta2) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie3 = e**(theta3) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie4 = e**(theta4) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))


prob_s1_5 = (pie1*poisson.pmf(5,lambda1))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
prob_s2_5 = (pie2*poisson.pmf(5,lambda2))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
prob_s3_5 = (pie3*poisson.pmf(5,lambda3))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
prob_s4_5 = (pie4*poisson.pmf(5,lambda4))/(pie1*poisson.pmf(5,lambda1) + pie2*poisson.pmf(5,lambda2) + pie3*poisson.pmf
print("Expected purchases next 8 weeks if past week = 5 packs is {}".format(8*(lambda1*(prob_s1_5) + lambda2*(prob_s2_5
maxi = max(prob_s1_5,prob_s2_5,prob_s3_5,prob_s4_5)
print("Expected purchases next 8 weeks if past week = 5 packs is, if customer is classified into segment1(since p(s=2,x
```

```
Expected purchases next 8 weeks if past week = 5 packs is 33.67181197102352
Expected purchases next 8 weeks if past week = 5 packs is, if customer is classified into segment1(since p(s=2,x=5) &
p(s=3,x=5) & p(s=1,x=5) < p(s=4,x=5)) is 24.015420513442717
```

For question 3, b)

```
#2segment
lambda2 = 1.802152486641123
lambda1 = 9.120665766225097
pie1 = 0.2991217614346511
pie2 = 0.7008782385653489

prob_s1_9 = (pie1*poisson.pmf(9,lambda1))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2))
prob_s2_9 = (pie2*poisson.pmf(9,lambda2))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2))
print("Expected purchases next 8 weeks if past week = 9 packs is {}".format(8*(lambda1*(prob_s1_9) + lambda2*(prob_s2_9
maxi = max(prob_s1_9,prob_s2_9)
print("Expected purchases next 8 weeks if past week = 9 packs is, if customer is classified into segment2(since p(s=2,x
```

```
Expected purchases next 8 weeks if past week = 5 packs is 72.87051083131314
Expected purchases next 8 weeks if past week = 5 packs is, if customer is classified into segment2(since p(s=2,x=9) <
p(s=1,x=9)) is 72.96532612980077
```

3-segment prediction

```
lambda1 = 3.483321426064036
lambda2 = 11.2158213224157
lambda3 = 0.29055258688349966
theta1 = 0.6744334195584446
theta2 = -0.43040650590390545
theta3 = 0.0
pie1 = e**(theta1) / ((e**theta1) + (e**theta2) + (e**theta3))
pie2 = e**(theta2) / ((e**theta1) + (e**theta2) + (e**theta3))
pie3 = e**(theta3) / ((e**theta1) + (e**theta2) + (e**theta3))

prob_s1_9 = (pie1*poisson.pmf(9,lambda1))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
prob_s2_9 = (pie2*poisson.pmf(9,lambda2))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
prob_s3_9 = (pie3*poisson.pmf(9,lambda3))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
print("Expected purchases next 8 weeks if past week = 9 packs is {}".format(8*(lambda1*(prob_s1_9) + lambda2*(prob_s2_9
maxi = max(prob_s1_9,prob_s2_9,prob_s3_9)
print("Expected purchases next 8 weeks if past week = 9 packs is, if customer is classified into segment1(since prob(s=
```

```
Expected purchases next 8 weeks if past week = 5 packs is 80.06350112030917
Expected purchases next 8 weeks if past week = 5 packs is, if customer is classified into segment1(since prob(s=1,x=
9) & prob(s=3,x=9) < prob(s=2,x=9)) is 89.7265705793256
```

4-segment prediction

```
lambda1 = 7.417654014502218
lambda2 = 0.2047757185785757
lambda3 = 12.872409491251577
lambda4 = 3.0019275641803396
theta1 = -1.2001167375208144
theta2 = -0.7220301738582039
theta3 = -1.5980245259938597
theta4 = 0.0
pie1 = e**(theta1) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie2 = e**(theta2) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie3 = e**(theta3) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))
pie4 = e**(theta4) / ((e**theta1) + (e**theta2) + (e**theta3) + (e**theta4))


prob_s1_9 = (pie1*poisson.pmf(9,lambda1))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
prob_s2_9 = (pie2*poisson.pmf(9,lambda2))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
prob_s3_9 = (pie3*poisson.pmf(9,lambda3))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
prob_s4_9 = (pie4*poisson.pmf(9,lambda4))/(pie1*poisson.pmf(9,lambda1) + pie2*poisson.pmf(9,lambda2) + pie3*poisson.pmf
print("Expected purchases next 8 weeks if past week = 9 packs is {}".format(8*(lambda1*(prob_s1_9) + lambda2*(prob_s2_9
maxi = max(prob_s1_5,prob_s2_5,prob_s3_5,prob_s4_5)
print("Expected purchases next 8 weeks if past week = 9 packs is, if customer is classified into segment1(since p(s=2,x
```

Part II: Analysis of New Data

Question 1:

Estimate all relevant parameters for Poisson regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood. What are the managerial takeaways — which customer characteristics seem to be important?

You may find the code for the Poisson Regression Model below:

```
df=pd.read_csv('articles.csv')
k=np.array(df[['articles']])
a=np.array(df[['female']])
b=np.array(df[['married']])
c=np.array(df[['kids']])
d=np.array(df[['prestige']])
f=np.array(df[['menpubs']])

def PR(params,a,b,c,d,f,k):
    lmbda0, beta1, beta2, beta3, beta4, beta5 = params
    p = 0
    pt1 = 0
    pt2 = 0
    pt3 = 0
    for i in range(len(a)):
        lmbda= lmbda0*math.exp(beta1*a[i]+beta2*b[i]+beta3*c[i]+beta4*d[i]+beta5*f[i])
        fact = float(math.factorial(float(k[i])))## take the log before
        pt1 = pt1 + k[i]*np.log(lmbda)
        pt2 = pt2 + lmbda
        pt3 = pt3 + np.log(fact)
    p = pt1-pt2-pt3
    return (-1)*p
```

```
lmbda0=0.0001
beta1=0.0001
beta2=0.0001
beta3=0.0001
beta4=0.0001
beta5=0.0001
params=lmbda0,beta1,beta2,beta3,beta4,beta5
solPR1 = minimize(
    PR,
    args = (a,b,c,d,f,k),
    x0 = np.array(params),
    bounds=[(0.00001, None), (None,None), (None,None), (None,None), (None,None), (None,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)

lmbda0 = solPR1.x[0]
beta1=solPR1.x[1]
beta2=solPR1.x[2]
beta3=solPR1.x[3]
beta4=solPR1.x[4]
beta4=solPR1.x[5]
ll_PR1=solPR1.fun[0]
print("Lambda0: {}".format(lmbda0))
print("Beta1: {}".format(beta1))
print("Beta2: {}".format(beta2))
print("Beta3: {}".format(beta3))
print("Beta4: {}".format(beta4))
print("Beta5: {}".format(beta5))
print("Log Likelihood: {}".format(ll_PR1*(-1)))
```

The corresponding parameters and log-likelihood are as follows:

Model	Poisson Regression model
Log Likelihood	-1651.0565708515671
Lambda 0	1.3569756192417102
Beta1 (female)	-0.2239520330683631
Beta2 (married)	0.1549662615389069
Beta3 (kids)	-0.18508761154450679
Beta4 (prestige)	0.02553308422651582
Beta5 (menpubs)	0.0001

Managerial Takeaways:

Through the Poisson regression, we are able to better understand the observed heterogeneity in this dataset. For instance, we see that some features will predict a lower number of articles for a PhD candidate, such as:

- being a woman
- having kids

Alternatively, some features are more positively correlated with the number of published articles, and predict a higher number of articles published by a PhD. For instance, features like:

- being married
- coming from a more prestigious department
- having a mentor who has published an article in the past 3 years

increase are more positively correlated, with marriage being the variable with the most positive predictor of number of articles. We would also like to note that while all betas changed from the initial value assigned to them before being ran through the optimization function, only beta5 (menpubs) remained unchanged. It seems that, in this Poisson regression, the number of mentor published articles does not have a major effect when predicting the number of articles of a PhD candidate.

Question 2:

Estimate all relevant parameters for NBD Regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood. What are the managerial takeaways — which customer characteristics seem to be important?

You may find the code for the NBD Regression Model below:

```
def NBDR(params,a,b,c,d,f,k):
    alpha, n, beta1, beta2, beta3, beta4, beta5 = params
    p = 0
    pt1 = 0
    pt2 = 0
    pt3 = 0
    for i in range(len(a)):
        pt1 = np.log(math.gamma(n+int(k[i]))) -(np.log(math.gamma(n)) + np.log(factorial(int(k[i]))))
        denom = math.exp(beta1*a[i]+beta2*b[i]+beta3*c[i]+beta4*d[i]+beta5*f[i])
        pt2 = n*(np.log(alpha) - np.log(alpha+denom))
        pt3 = int(k[i])*(np.log(denom) - np.log(alpha+denom))
        p += pt1 + pt2 + pt3
    return -1*p

alpha=0.01
n=0.0001
beta1=0.0001
beta2=0.5
beta3=0.02
beta4=0.3
beta5=0.1

params=alpha,n,beta1,beta2,beta3,beta4,beta5
solnbdr1 = minimize(
    NBDR,
    args = (a,b,c,d,f,k),
    x0 = np.array(params),
    bounds=[(0.000001,None),(0.000001,None),(None,None),(None,None),(None,None),(None,None),(None,None)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	NBD Regression model
Log Likelihood	-1560.9585966009363
alpha	1.7498866921285627
n	2.2649021059591434
Beta1 (female)	-0.2157650356969458
Beta2 (married)	0.1492626869924588
Beta3 (kids)	-0.17626768965454373
Beta4 (prestige)	0.014831939706369512
Beta5 (menpubs)	0.029098848354927986

Managerial Takeaways:

The NBD Regression returned a significantly higher log likelihood than the Poisson model. This points to the necessity of accounting for unobserved heterogeneity, which was not included in the Poisson Regression. Through the NBD regression, we are able to better understand both the unobserved and observed heterogeneity in this dataset. Focusing on observed heterogeneity, we see that some features will predict a lower number of articles for a PhD candidate, such as:

- being a woman
- having kids

Alternatively, some features are more positively correlated with the number of published articles, and predict a higher number of articles published by a PhD. For instance, features like:

- being married
- coming from a more prestigious department
- having a mentor who has published an article in the past 3 years

increase are more positively correlated, with marriage being the variable with the most positive predictor of number of articles. We did note that most of the beta coefficients are lower in the NBD model than in the Poisson model. We can hypothesize that, due to the fact that some of the variation is now associated with an unobserved heterogeneity, the observed beta coefficients will account for less of the explanation of variation in the data.

Question 3:

In this question, you will apply the ideas learned in this course to build a model that you have not seen before – the Zero Inflated NBD Regression. First, recall that zero inflated models view 0s as coming from 2 sources - (i) from a fraction π who is 0 "by type" (in the context of this problem, these are candidates who will never publish), and (ii) from the remaining fraction $(1 - \pi)$ who are likely to eventually become nonzero (these are candidates who will publish at some point, but have not done so yet). You can assume that the candidates in the latter group are distributed as a negative binomial (making the NBD regression appropriate for them). Explain the logic used in developing the model in detail. (hint: you do not need anything beyond what you have learned in the class to do this.) Report your code, the estimated parameters and the maximum value of the log-likelihood. What are the managerial takeaways — which customer characteristics seem to be important?

Please find the code for the Zero Inflated NBD Regression below:

```
#1. we utilized our NBD Regression function and altered it to include zero inflated cases
def NBDRZI(params,a,b,c,d,f,k):
    alpha, n, beta1, beta2, beta3, beta4, beta5, pi = params
    p = 0
    for i in range(len(a)):
        pt1 = np.log(math.gamma(n+int(k[i]))) -(np.log(math.gamma(n)) + np.log(factorial(int(k[i]))))
        denom = math.exp(beta1*a[i]+beta2*b[i]+beta3*c[i]+beta4*d[i]+beta5*f[i])
        pt2 = n*(np.log(alpha) - np.log(alpha+denom))
        pt3 = int(k[i])*(np.log(denom) - np.log(alpha+denom))
        pt4 = np.log(1-pi)
        if k[i]==0: #this will better predict the number of 0's in the model
            pt1 = math.gamma(n+int(k[i]))/math.gamma(n)*factorial(int(k[i]))
            pt2 = (alpha/(alpha+denom))**n
            pt3 = (denom/(alpha+denom))**int(k[i])
            p += np.log(pt1*pt2*pt3*(1-pi)+pi)
        else:
            p += pt1 + pt2 + pt3 + pt4
    return -1*p

#2. we set a few guesses for each of the parameters
alpha=0.07
n=0.03
beta1=0.0001
beta2=0.6
beta3=0.4
beta4=0.03
beta5=0.005
pi=0.01 #pi is a probability, so we set it between 0 and 1, noninclusive
params=alpha,n,beta1,beta2,beta3,beta4,beta5,pi

#3. we optimize the parameters to maximize ll
solnbdrzi = minimize(
    NBDRZI,
    args = (a,b,c,d,f,k),
    x0 = np.array(params),
    bounds=[(0.000001,None),(0.000001,None),(None,None),(None,None),(None,None),(None,None),(None,None),(0.0000001,1)],
    tol=1e-10,
    options={'ftol' : 1e-8},
)
```

The corresponding parameters and log-likelihood are as follows:

Model	ZI NBD Regression model
Log Likelihood	-1560.9584097116997
alpha	1.750820546616006
n	2.264466660799976
pi	0.0000001
Beta1 (female)	-0.21654521223665577
Beta2 (married)	0.1497872530672166
Beta3 (kids)	-0.17602209416685324
Beta4 (prestige)	0.015055583169360948
Beta5 (menpubs)	0.029078581162349137

Managerial Takeaways:

Through the Zero Inflated NBD regression, we are able to better understand the observed heterogeneity in this dataset. For instance, we see that some features will predict a lower number of articles for a PhD candidate, such as:

- being a woman
- having kids

Alternatively, some features are more positively correlated with the number of published articles, and predict a higher number of articles published by a PhD. For instance, features like:

- being married
- coming from a more prestigious department
- having a mentor who has published an article in the past 3 years

increase are more positively correlated, with marriage being the variable with the most positive predictor of number of articles. We would like to note that, in our ZI NBD Regression model, pi is taking the smallest possible value it can take, depending on the bounds assigned. This points to the fact that the number of people who will never publish is really low, and most of these candidates are likely to publish an article—they just have not done so yet.

Question 4:

Evaluate the models developed; explain which of them is best, and why. Are there any significant differences among the results from these models? If so, what exactly are these differences? Discuss what you believe could be causing the differences.

You may find the summary of our regressions and evaluations below:

Q#	Model	LL	# of params	AIC	BIC
Q1	Poisson	-1651.05	5	3312.1	3336.21
Q2	NBD Regression	-1560.95	6	3133.9	3136.83
Q3	ZI NBD Regression	-1560.95	8	3137.9	3176.47

Based on the above summary, we can see that the NBD Regression outperforms the Poisson regression and ZI-NBD regression because it possesses the highest log-likelihood ratio, the lowest BIC and the second low AIC.

The log-likelihood significantly improved in the NBD regression as well as ZI-NBD regression (-1560.95) if compare with the Poisson regression (-1651.05).

We can see that the log-likelihood ratio of NBD regression and ZI-NBD regression are higher than that of the Poisson regression. We believe that's because the parameters n and alpha from NBD regression help us further capture unobserved heterogeneity from the original records. It not only recognizes the existence of observed heterogeneity but captures unobserved heterogeneity as well; therefore, it could have resulted in a higher log-likelihood.

Furthermore, we thought it would be important to include the betas associated with each variable for each of the regressions ran above. You may find the summary in the table below:

Variables	Poisson Regression	NBD Regression	ZI NBD Regression
Female	-0.2239520330683631	-0.2157650356969458	-0.21654521223665577
Married	0.1549662615389069	0.1492626869924588	0.1497872530672166
Kids	-0.1850876115445067	-0.17626768965454373	-0.17602209416685324
Prestige	0.02553308422651582	0.014831939706369512	0.015055583169360948
Menpubs	0.0001	0.029098848354927986	0.029078581162349137

We believe our regressions point to a few consistent takeaways. First, the PhD candidates' estimation of articles published was consistently penalized when the candidate was either a woman or had kids. Second, marriage was strongly associated to a higher number of published articles in all three of our regressions. Finally, Prestige and Mentor publishing had only a small effect when predicting the number of articles a PhD candidate would have.

Managerial Takeaways:

As with project 01, briefly summarize what you learned from project 02. Remember – this is an open-ended question, so please include anything you found worthwhile.

Part 1 - summary and findings:

While working on part 1, we were able to replicate the findings for the candy dataset we found in the classroom. This exercise allowed us to better understand how models like the Zero-Inflated NBD model and the Finite Mixture Models are derived from the NBD and Poisson models, since we reused the original functions we had developed in project 1 and modified them to fit our desired outcomes. The progression of the project allowed us to make more definitive connections between these models.

Furthermore, the log likelihood and evaluation measures (AIC/BIC) furthered our understanding of these models – which model fits the data better, what intuition best explains the behavior behind the data, and ultimately, which model to implement.

Part 2 – summary and findings:

Part 2 of this project allowed us to venture into a completely new dataset. We were able to truly reinforce our understanding of the different models when trying to build the Zero Inflated NBD Regression model, as it was not introduced as such during our classwork. Furthermore, the implementation of such complex models (up to 8 variables, in the case of ZI NBD Regression) allowed us to see that the model often hit a plateau in terms of its log likelihood improvement. It was through this context that we found measures like BIC especially helpful, as a way to prevent overfitting.

In terms of the content of part 2 of this project, we found the substance quite interesting. We found that although the overall log-likelihood results from the Poisson Regression, NBD Regression and ZI NBD Regression models were different, the effects of the variables were (at least, directionally) consistent across the board: being a woman and having kids was often associated with fewer articles published, while being married consistently increased this number.