THE UNIVERSITY OF TEXAS AT DALLAS

NAVEEN JINDAL SCHOOL OF MANAGEMENT

COURSE NAME: MODELING FOR BUSINESS ANALYTICS

COURSE CODE: BUAN 6383/MIS 6386 001

GROUP NUMBER: 9

Project: 02

OBJECTIVE: Learn to build advanced customized models

GROUP MEMBERS:

Anushree Sanagi Jagadeesh

Ruchika Gupta

Chetan Kulkarni

Poorvik Gambheer

Part I: Replicating Models from Class

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):

(a) the Poisson model, (b) the NBD model, (c) the Zero Inflated NBD model, and (d) Finite Mixture models for 2, 3, and 4 segments. 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,

Below is the code used for the model.

```
In [8]:  poisson_pmf = lambda k,l: k*np.log(l)-l*(np.log(e))-np.log(factorial(k))
```

```
In [9]:
def LL(params,inputs):
    lambda0=params
    e=inputs['Packs']
    p=inputs['People']
    sum=0
    for i in range(len(inputs)):
        sum+=poisson_pmf(e[i],lambda0)*p[i]
    return sum
```

```
In [10]: def NLL(params, inputs):
             return(-(LL(params, inputs)))
```

```
In [11]: final=minimize(NLL,
                  args=inputs,
                  x0=params,
                  bounds=[(0.000001, None)],
                 tol=1e-10,
                 options={'ftol' : 1e-8},)
```

```
In [12]: final
```

```
Out[12]:      fun: array([1544.99639045])
         hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
              jac: array([-2.27373677e-05])
          message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
             nfev: 24
              nit: 10
             njev: 12
           status: 0
          success: True
                x: array([3.99122805])
```

Below are the results:

The Value for Lambda is 3.9912. The Value of Maximum Log likelihood is -1544.9964

(b) The NBD Model

Below is the code:

(b) the NBD model,

```
params = np.array([0.1,0.1])
```

```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t
```

```
def NBDLL(params,inputs):
    e=inputs['Packs']
    p=inputs['People']
    n=params[0]
    alpha=params[1]
    sum=0
    for i in range(len(inputs)):
        sum+=pmf_nbd(alpha,n,e[i],1)*p[i]
    return sum
```

```
def NLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None)])
```

```
final
```

```
22]:      fun: 1140.023746185597
     hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
          jac: array([ 0.00020464, -0.00111413])
      message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
         nfev: 39
          nit: 11
         njev: 13
       status: 0
      success: True
            x: array([0.99765785, 0.24996243])
```

Below are the results:

The Value for n is 0.99766 and Alpha is 0.24996 .The Value of Maximum Log likelihood is -1140.0237

(c) The Zero Inflated NBD model

Code:

```
params = np.array([0.1,0.1,0.1])
```

```
pmf_zinbd=lambda a,n,k,t : ((gamma(n+k))/((gamma(n))*factorial(k)))*((a/(a+t))**n)*((t/(a+t))**k)
```

```
def ZINBDLL(params,inputs):
    e=inputs['Packs']
    p=inputs['People']
    pie=params[0]
    alpha=params[1]
    n=params[2]
    sumi=0
    ll = 0
    for i in range(len(inputs)):
        if i == 0:
            sumi = pie + ((1-pie) * (pmf_zinbd(alpha,n,e[i],1)))
        else:
            sumi = ((1-pie) * (pmf_zinbd(alpha,n,e[i],1)))
        ll += p[i] * np.log(sumi)

    return ll
```

```
def NLL(params, inputs):
    return(-(ZINBDLL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, 0.999999),(0.000001, None),(0.000001, None)],
         tol=1e-10,
         options={'ftol' : 1e-8},)
```

```
final
```

```
]:       fun: 1136.1656408391289
    hess_inv: <3x3 LbfgsInvHessProduct with dtype=float64>
         jac: array([ 0.00500222,  0.00034106, -0.00045475])
     message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
        nfev: 72
         nit: 16
        njev: 18
      status: 0
     success: True
           x: array([0.11310781, 0.334189  , 1.5039286 ])
```

Results:

The Value for pie is 0.1131 and Alpha is 0.3342 . The Value of Maximum Log likelihood is -1136.1656

(d) Finite Mixture models for 2, 3, and 4 segments

Finite Mixture models for 2 segments:

(d) Finite Mixture models for 2 segments.

```
params = np.array([0.1,0.1,0.1])
```

```
poisson_pmf1 = lambda l1,k : (l1**k)*(e**(-l1))/factorial(k)
```

```
poisson_pmf2 = lambda l2,k : (l2**k)*(e**(-l2))/factorial(k)
```

```
def ZINBDLL(params,inputs):
    e=inputs['Packs']
    p=inputs['People']
    pie=params[0]
    l1=params[1]
    l2=params[2]
    sumi=0
    ll = 0
    for i in range(len(inputs)):
            sumi = (pie * (poisson_pmf1(l1,e[i]))) + ((1-pie) * (poisson_pmf2(l2,e[i])))
            ll += p[i] * np.log(sumi)

    return ll
```

```
def NLL(params, inputs):
    return(-(ZINBDLL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, 0.999999),(0.000001, None),(0.000001, None)],
         tol=1e-10,
         options={'ftol' : 1e-8},)
```

Results:

The Value for pie is 0.70088 and

Lambda1 is 1.80215 and

Lambda2 is 9.12069 .

The Value of Maximum Log likelihood is -1188.8328271734968

Finite Mixture models for 3 segments:

(d) Finite Mixture models for 3 segments.

```
params = np.array([1,1,1,2,1])
```

```
pie = lambda theta1, theta2: ((e**theta1)/(e**theta1 + e**theta2 + 1)) + ((e**theta2)/(e**theta1 + e**theta2 + 1)) + (1/(e**t
```

```
pie_sum = lambda theta1, theta2: (e**theta1 + e**theta2 + 1)
```

```
poisson_pmf = lambda l1,l2,l3,theta1,theta2,k : ((l1**k)*(e**(-l1))/factorial(k))*(e**theta1/pie_sum(theta1,theta2)) + ((l2**
```

```
def SEG3LL(params,inputs):
    e=inputs['Packs']
    p=inputs['People']
    l1=params[0]
    l2=params[1]
    l3=params[2]
    theta1=params[3]
    theta2=params[4]
    sumi=0
    ll = 0
    for i in range(len(inputs)):
        sumi = poisson_pmf(l1,l2,l3,theta1,theta2,e[i])
        ll += p[i] * np.log(sumi)
    return ll
```

```
def NLL(params, inputs):
    return(-(SEG3LL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None),(0.000001, None),(None, None),(None, None)],
         tol=1e-10,
         options={'ftol' : 1e-8},)
```

```
final
```

The Value for lambda1 is 11.2158
Lambda2 is 3.4833
Lambda3 is 0.2905
theta1 is -0.4304
theta2 is 0.6744321103382401
The Value of Maximum Log likelihood is -1132.0429842731146

Finite Mixture models for 4 segments:

(d) Finite Mixture models for 4 segments.

```
params = np.array([1,1,1,1,1,2,3])
```

```
pie = lambda theta1, theta2,theta3: ((e**theta1)/(e**theta1 + e**theta2 + e**theta3 + 1)) + ((e**theta2)/(e**theta1 + e**thet
```

```
pie_sum = lambda theta1, theta2, theta3: (e**theta1 + e**theta2 + e**theta3 + 1)
```

```
poisson_pmf = lambda l1,l2,l3,l4,theta1,theta2,theta3,k : ((l1**k)*(e**(-l1))/factorial(k))*(e**theta1/pie_sum(theta1,theta2,
```

```
def SEG3LL(params,inputs):
    e=inputs['Packs']
    p=inputs['People']
    l1=params[0]
    l2=params[1]
    l3=params[2]
    l4=params[3]
    theta1=params[4]
    theta2=params[5]
    theta3=params[6]
    sumi=0
    ll = 0
    for i in range(len(inputs)):
        sumi = poisson_pmf(l1,l2,l3,l4,theta1,theta2,theta3,e[i])
        ll += p[i] * np.log(sumi)
    return ll
```

```
def NLL(params, inputs):
    return(-(SEG3LL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None),(0.000001, None),(0.000001, None),(None, None),(None, None),(None, None)],
         tol=1e-10,
         options={'ftol' : 1e-8},)
```

```
final
```

Results:

The Value for lambda1 is 7.4191

Lambda2 is 0.2047

Lambda3 is 12.8726

Lambda4 is 3.0020

theta1 is -1.2001

theta2 is -0.7220

The Value of Maximum Log likelihood is -1130.0705

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.

Among the models developed so far, below is a table comparing the AIC, BIC and Log Likelihood.

Model	n	k	LL	AIC	BIC
Poisson Model	21	1	-1544.996	3091.993	3093.037
NBD Model	21	2	-1140.024	2284.047	2286.137
Zero Inflated NBD model	21	3	-1136.166	2278.331	2281.465
Finite Mixture models for 2 segments	21	3	-1188.833	2383.666	2386.799
Finite Mixture models for 3 segments	21	5	-1132.043	2274.086	2279.309
Finite Mixture models for 4 segments	21	7	-1130.071	2274.141	2281.453

The Finite Mixture model with 3 segments has the least AIC, BIC values and the highest Log Likelihood values so this is the best model among the ones developed so far. The difference between the models is the variability in the value of k. As can be concluded from the results above, Log Likelihood value gets better (maximize) with the increase in the value of k.

Additionally, the AIC and BIC values are minimum for Finite Mixture model with 3 segments which makes this model better than others.

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

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

We are using the below formula to answer the above question

$$p(s|x=k) = \frac{\pi_s \times p(x=k|\lambda_s)}{\sum_{i=1}^{S} \pi_i \times p(x=k|\lambda_i)}$$

Based on the 2,3 and 4 segment finite mixture model, the model with 3 segments gives the best results.

Below are the parameters obtained from the model

The Value for lambda1 is 11.2158

Lambda2 is 3.4833

Lambda3 is 0.29055

theta1 is -0.43040617532093817

theta2 is 0.6744321103382401

Value of pie1= 0.1514010119410262

Value of pie2= 0.24419618755327793

Value of pie3= 0.10166792969361256

Value of pie4= 0.5027348708120832The Value of Maximum Log likelihood is -1132.0429842731146

a) Below are the steps taken to get the values of expected purchases for a customer that purchased 5 packs last week.

```
pie1= e**(final.x[3])/pie_sum(final.x[3],final.x[4])
pie2= e**(final.x[4])/pie_sum(final.x[3],final.x[4])
pie3= 1/pie_sum(final.x[3],final.x[4])
lambda1=final.x[0]
lambda2=final.x[1]
lambda3=final.x[2]
print("Value of pie1=",pie1)
print("Value of pie2=",pie2)
print("Value of pie3=",pie3)
print("Value of lambda1=",lambda1)
print("Value of lambda2=",lambda2)
print("Value of lambda3=",lambda3)
```

```
Value of pie1= 0.17996557357392923
Value of pie2= 0.5432686018493437
Value of pie3= 0.2767658245767271
Value of lambda1= 11.21580038437127
Value of lambda2= 3.483318571593005
Value of lambda3= 0.29055239542745726
```

```
poisson_pmf_one_var = lambda k,l: k*np.log(l)-l*(np.log(e))-np.log(factorial(k))
```

```
#calculating the denominator for customer who purchased 5 packs in segment 1.
Denominator=(poisson_pmf_one_var(5,lambda1)*pie1)+(poisson_pmf_one_var(5,lambda2)*pie2)+(poisson_pmf_one_var(5,lambda3)*pie3
```

```
Numerator1=(poisson_pmf_one_var(5,lambda1)*pie1)
Numerator2=(poisson_pmf_one_var(5,lambda2)*pie2)
Numerator3=(poisson_pmf_one_var(5,lambda3)*pie3)
```

```
#p(s=1|x=5)
P_Seg_1=Numerator1/Denominator
#p(s=2|x=5)
P_Seg_2=Numerator2/Denominator
#p(s=3|x=5)
P_Seg_3=Numerator3/Denominator
```

```
#expected Purchases in one week based on the fact that customer has purchased 5 packs last week
Exp_1week=lambda1*P_Seg_1+lambda2*P_Seg_2+lambda3*P_Seg_3
```

```
#Expected Purchases in 8 weeks
Exp_8weeks=Exp_1week*8
```

```
print("Expected purchases with 3 segment finite mixture model where customer purchased 5 packs last week is ", Exp_8weeks)
```

```
Expected purchases with 3 segment finite mixture model where customer purchased 5 packs last week is  20.55919703507159
```

Expected purchases with 3 segment finite mixture model where customer purchased 5 packs last week is 20.559

b) Below is the code for the customer who purchased 9 packs last week.

A customer who purchased 9 packs in the past week.

```
#calculating the denominator for customer who purchased 9 packs in segment 1.
Denominator=(poisson_pmf_one_var(9,lambda1)*pie1)+(poisson_pmf_one_var(9,lambda2)*pie2)+(poisson_pmf_one_var(9,lambda3)*pie3)
```

```
Numerator1=(poisson_pmf_one_var(9,lambda1)*pie1)
Numerator2=(poisson_pmf_one_var(9,lambda2)*pie2)
Numerator3=(poisson_pmf_one_var(9,lambda3)*pie3)
```

```
#p(s=1|x=9)
P_Seg_1=Numerator1/Denominator
#p(s=2|x=9)
P_Seg_2=Numerator2/Denominator
#p(s=3|x=9)
P_Seg_3=Numerator3/Denominator
```

```
#expected Purchases in one week based on the fact that customer has purchased 9 packs last week
Exp_1week=lambda1*P_Seg_1+lambda2*P_Seg_2+lambda3*P_Seg_3
```

```
#Expected Purchases in 8 weeks
Exp_8weeks=Exp_1week*8
```

```
print("Expected purchases with 3 segment finite mixture model where customer purchased 9 packs last week is ", Exp_8weeks)
```

```
Expected purchases with 3 segment finite mixture model where customer purchased 9 packs last week is  13.050042128951457
```

Expected purchases with 3 segment finite mixture model where customer purchased 9 packs last week is 13.050

Part II: Analysis of New Data

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?

Please find the full code on the file. Below is a snapshot:

```
In [106]: params = np.array([0.01,0.01,0.1,0.01,0.01,0.01])

In [107]: params[1:len(params)]

Out[107]: array([0.01, 0.1 , 0.01, 0.01, 0.01])

In [108]: def LL(params,inputs):
              k=inputs['articles'].tolist()
              kc1=inputs.drop(['articles'],axis=1)
              a=kc1.to_numpy().tolist()
              beta=params[1:len(params)]
              lambda0=params[0]
              lambdai=0
              sum2=0
              sum1=0

              for i in range(len(a)):
                  sum1=0
                  for j in range(len(a[i])):
                      sum1=sum1+beta[j]*a[i][j]
                  lambdai=lambda0*np.exp(sum1)
                  sum2=sum2+poisson_pmf(k[i],lambdai)
              return sum2

In [109]: def NegLL(params, inputs):
              return(-(LL(params, inputs)))

In [110]: final=minimize(NegLL,
                  args=inputs,
                  x0=params,
                  bounds=[(0.000001, None),(None, None), (None, None),(None, None), (None, None),(None, None)])

In [111]: print("The Value for lambda is ",final.x[0],"Beta parameters are ",final.x[1:len(params)],".The Value of Maximum Log likelihc
```

Parameters	Values
Lambda	1.3560
Maximum Log Likelihood	-1651.0563
Beta1	-0.2245
Beta2	0.1554
Beta3	-0.1849
Beta4	0.01283
Beta5	0.0255

After performing Poisson Regression on the dataset, we can imply that

- The most relevant feature is “married”. Married doctoral candidates publish more articles. All married individuals, publish 0.1554 times more than unmarried individuals.
- It is also evident that females don’t publish as much as males. Having more kids also negatively impacts publishing articles. A unit increase in the number of kids will reduce the number of publications by 0.18 times.
- Prestige of the candidate’s department and the number of publications by the candidate’s mentor over the past 3 years have very low beta values that means they don’t significantly impact the number of publications.
- Poisson regression has not sufficiently tried to explain the change in the dependant variable largely.

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?

Please find the full code on the file. Below is a snapshot:

```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t
```

```
def NBDLL(params,inputs):
    k=inputs['articles'].tolist()
    kc1=inputs.drop(['articles'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params[2:len(params)]
    alpha=params[0]
    n=params[1]
    lambdai=0
    sum2=0
    sum1=0
    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        t=np.exp(sum1)
        sum2=sum2+pmf_nbd(alpha,n,k[i],t)
    return sum2
```

```
def NegLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

```
final=minimize(NegLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None),(None, None), (None, None), (None, None), (None, None),(None, None)])
```

```
print("The Value for n is ",final.x[1]," and Alpha is ",final.x[0])
print("The values for the beta parameters are",final.x[2:len(params)])
print("The Value of Maximum Log likelihood is ",-1*final.fun)
```

Parameters	Values
n	2.2642
alpha	1.7524
Maximum Log Likelihood	-1560.9583
Beta1	-0.2164
Beta2	0.1505
Beta3	-0.1764
Beta4	0.0152
Beta5	0.0291

After performing NBD Regression, we can imply the following:

- The most relevant feature is “married”. Married doctoral candidates publish more articles. All married individuals, publish 0.1505 times more than unmarried individuals.
- It is also evident that females don’t publish as many times as males. Having more kids also negatively impacts publishing articles. A unit increase in the number of kids will reduce the number of publications by 0.1764 times.
- Prestige of the candidate’s department and the number of publications by the candidate’s mentor over the past 3 years have very low beta values that means they don’t significantly impact the number of publications.
- NBD regression has better than log likelihood as compared to Poisson.

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.

Please find the full code on the file. Below is a snapshot:

```
In [66]:  def NBDLL(params,inputs):
              k=inputs['articles'].tolist()
              kc1=inputs.drop(['articles'],axis=1)
              a=kc1.to_numpy().tolist()
              beta=params[2:7]
              alpha=params[0]
              n=params[1]
              pie=params[2]
              sum2=0
              sum1=0
              ll=0
              for i in range(len(a)):
                  sum1=0
                  for j in range(len(a[i])):
                      sum1=sum1+beta[j]*a[i][j]
                  t=np.exp(sum1)
                  if a[i] == 0:
                      sum2 = sum2 + np.log(pie + ((1-pie) * (pmf_zinbd(alpha,n,k[i],t))))
                  else:
                      sum2 = sum2 + np.log((1-pie)) + (pmf_nbd(alpha,n,k[i],t))
              return sum2

In [67]:  def NegLL(params, inputs):
              return(-(NBDLL(params, inputs)))

In [68]:  NegLL(params,inputs)

Out[68]:  3254.9312645281493

In [69]:  final=minimize(NegLL,
                   args=inputs,
                   x0=params,
                   bounds=[(0.000001, None),(0.000001, None),(0.000001,0.999999),(None, None), (None, None), (None, None), (None, None)
                   tol=1e-10,
                   options={'ftol' : 1e-8})
```

Parameters	Values
n	2.2642
alpha	1.7524
Pi	-0.2164
Maximum Log Likelihood	-1560.9583
Beta1	-0.2164
Beta2	0.1505
Beta3	-0.1764
Beta4	0.0152
Beta5	0.0291

The logic used in the model basically implements the NBD Regression using the following formula

P(x=k|n,alpha) = pie + (1-pie)* (PMF function for NBD)

Where PMF function for NBD is

$$\frac{\Gamma(n+k)}{\Gamma(n)k!}\left(\frac{\alpha}{\alpha+e^{\boldsymbol{\beta}\mathbf{x_i}}}\right)^{n}\left(\frac{e^{\boldsymbol{\beta}\mathbf{x_i}}}{\alpha+e^{\boldsymbol{\beta}\mathbf{x_i}}}\right)^{k}$$

For x=0, the formula becomes pie + (1-pie)* (PMF function for NBD)

And for x >0 , the formula becomes (1-pie) * (PMF function for NBD)

After this we maximize the log of the output using the minimize function and generate the parameters n,alpha, pie and the beta’s

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.

Comparing the models we can summarize the below:

Part	Question No.	Model Name	Log Likelihood (LL)	Parameters (k)	Observations	AIC	BIC
2	1	**Poisson Regression**	-1651.05630	6	915	3314.113	3319.881
2	2	**NBD Regression**	-1560.95830	7	915	3135.917	3142.647
2	3	**Zero Inflated NBD Regression**	-1560.95830	8	915	3137.917	3145.608

- Among all the models, we tried to implement zero inflated NBD regression, In order to understand if the zeros are true zeros.
- **It seems that, the dataset has about 275 zeros which are less when compared to the size of dataset. The zero inflated NBD gives the same results as the normal NBD regression because, there may not be zero inflation in the data.**
- Among all the models, NBD regression works the best. With better log likelihood.
- **The differences between Poisson's and NBD Regressions comes from the fact that we also include the heterogeneity in the NBD regression which caused the Likelihood to increase. And since there was no zero inflation in the data, zero-inflated NBD did not really produce a different result compared to NBD.**
- **Additionally, it was observed that since the number of parameters are more in the Zero inflated NBD model, the AIC and BIC values are higher which indicates that the model is not the best one to use compared to the NDB model.**