Name: Akanksha Dewangan, Roll No.: MT19049

Assignment - 2

Ans 1-

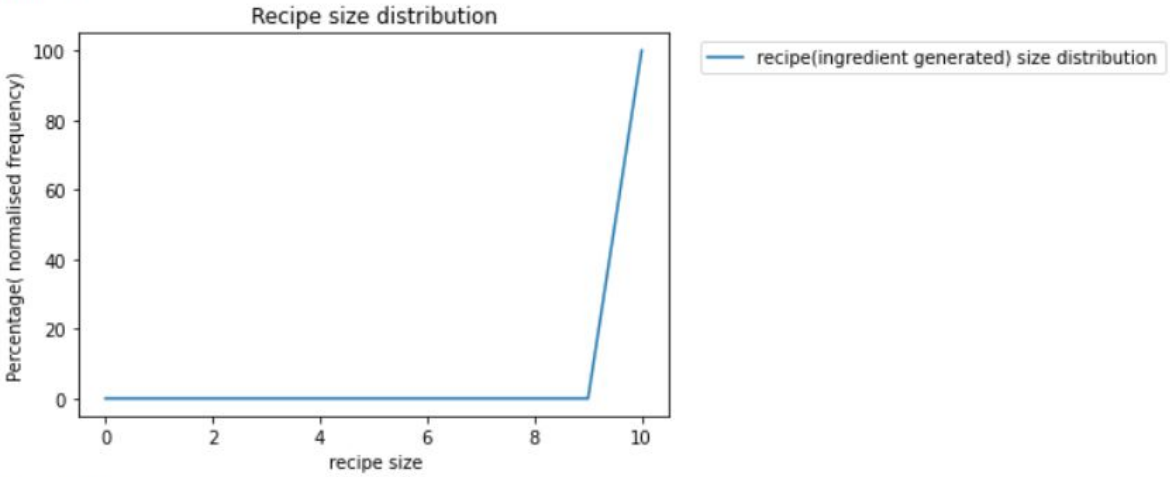
1. Start with Epoch =0 :
Number of Recipes Per Epochs = Round (Number of recipes/Number of Epochs)
-fitness value assigned to each ingredient are unique and assign as per uniform distribution.
EXPLANATION:
-EPOCH-
Epoch is defined as the time period with that time our cuisine is updated and the recipes modify after a certain time with (unit / times period) and to frame it in our algorithm by each epoch we make the required recipes we want in each round.

I had taken epoch i equal to 6 as assumption and for that in each epoch 6629 recipes get generated so by running in this way we reach 6629*6=39774 recipes and along with that we used up all the ingredient from nature basket and fill to in kitchen basket as shown in followings picture of epoch 0 to 5(all 6 epochs):
Rounds of epoch = (Number of recipes / Number of epochs), this much rounds required to generate recipes.

Following For epoch=6:
In the following graph all the details of
-M ratio
-number of recipes(currently in primordial cuisine)
-current size of kitchen basket.
-current size of nature basket.
-Also calculator how : Rounds:(Number of recipes / Number of epochs).
That is how many rounds the inner loop for recipes generation will run to update primordial cuisine.

A.Epoch:0

```
M ratio:  0.1
In epoch:  0
 Number of recipes :  500
 Size of recipes : 10
 Kitchen bastket size is:  50
 nature basket :  6664
Epoch:  0
```



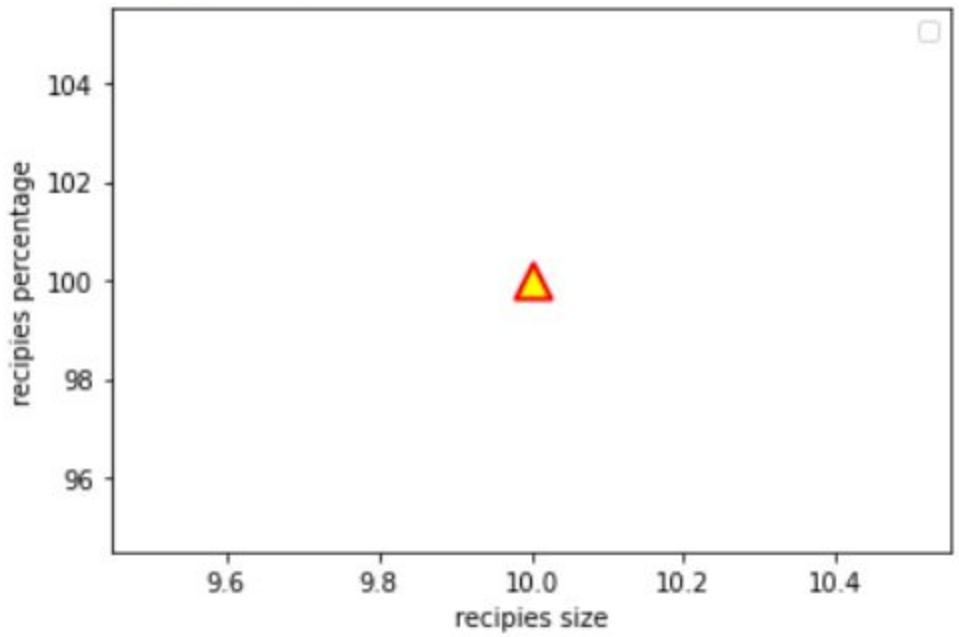

```
X:  1119
 total_number_recipies_generated_so_far:  6629
Nature basket:  5595
 kitchen basket:  1119
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Analysis:

Average Recipe size in epoch is 10, showing a yellow triangle.:

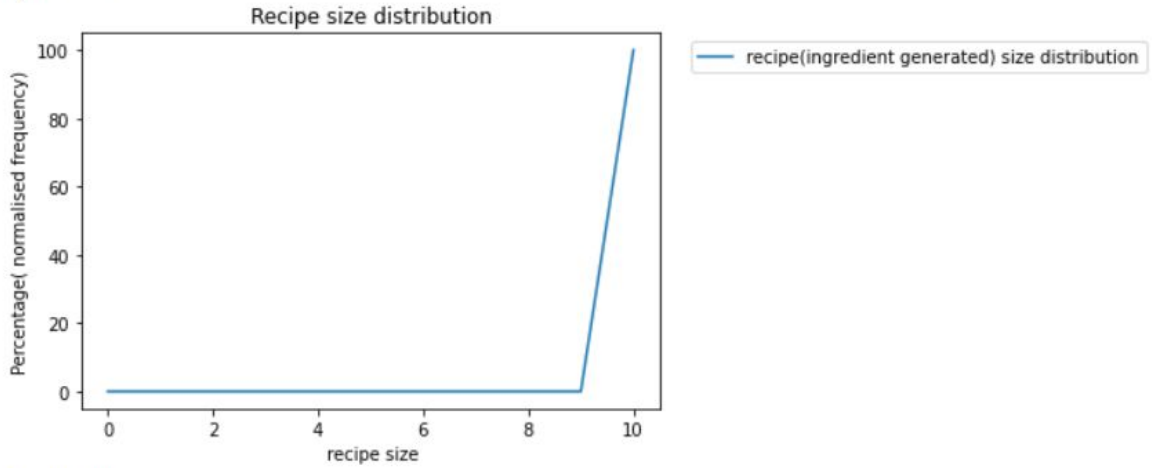
In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.

EPOCH: 0



B.Epoch 1-

```
M ratio:  0.18257464512971122
In epoch:  1
 Number of recipes :  6129
 Size of recipes : 10
 Kitchen bastket size is:  1119
 nature basket :  5595
Epoch:  1
```



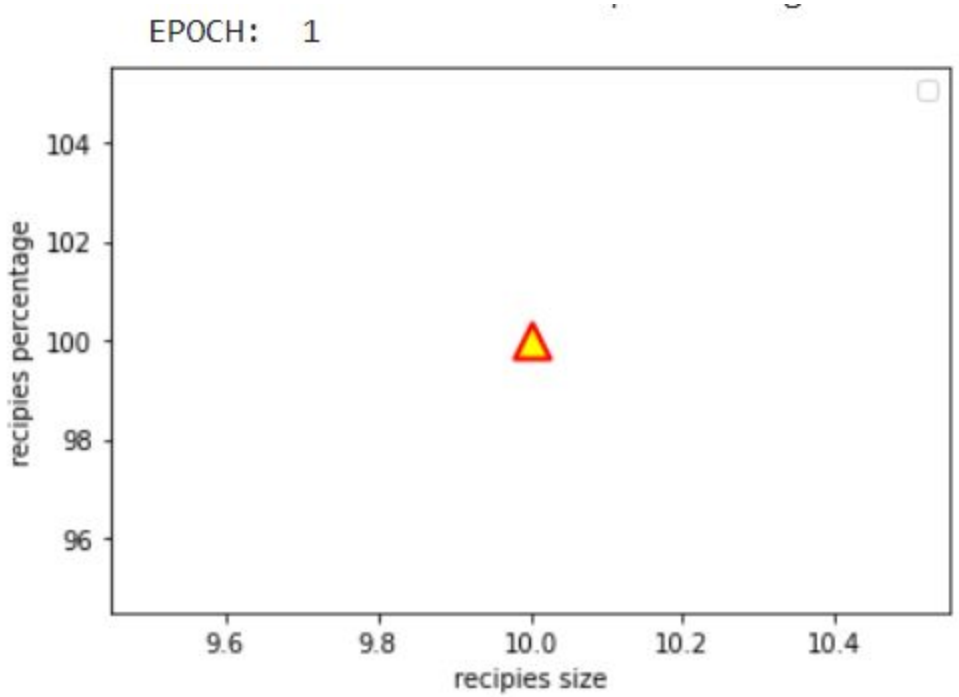

```
X:  2238
 total_number_recipies_generated_so_far:  13258
Nature basket:  4476
 kitchen basket:  2238
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Average Recipe size in epoch is 10, showing yellow triangle.

Analysis:

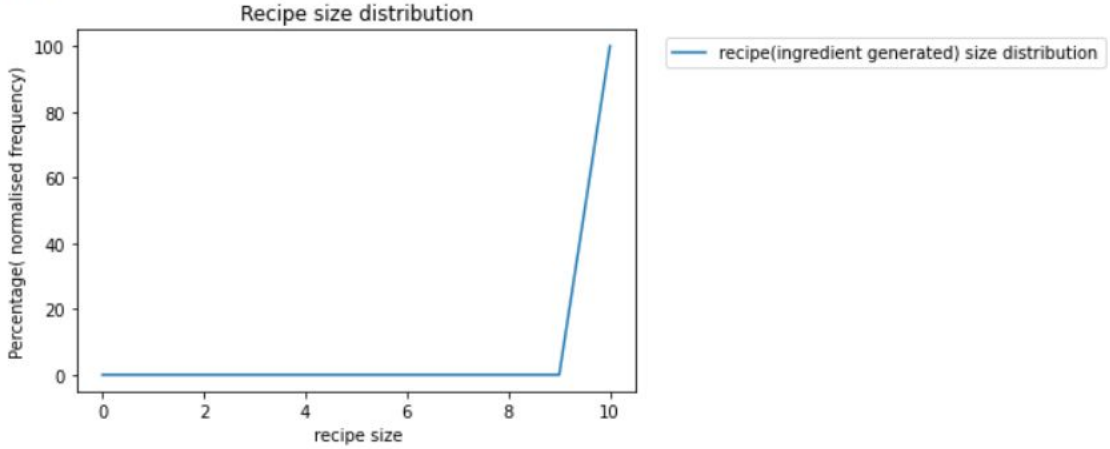
Average Recipe size in epoch is 10, showing a yellow triangle.:

In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.



C.Epoch 2-

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
M ratio:  0.16880374113742647
In epoch:  2
 Number of recipes :  13258
 Size of recipes : 10
 Kitchen bastket size is:  2238
 nature basket :  4476
Epoch:  2
```




```
X:  3357
 total_number_recipies_generated_so_far:  19887
Nature basket:  3357
 kitchen basket:  3357
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

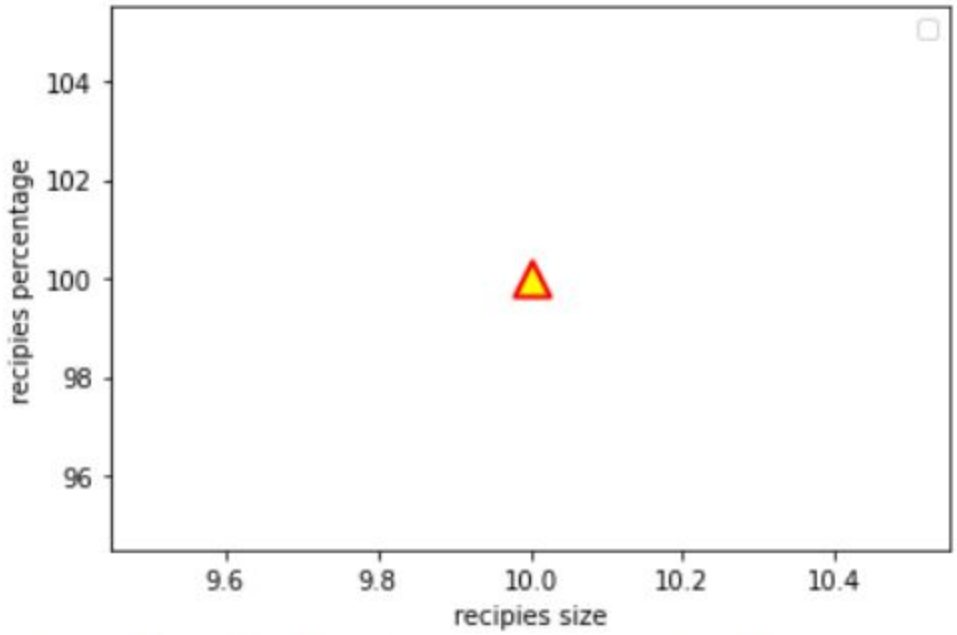
Average Recipe size in epoch is 10, showing a yellow triangle.

Analysis:

Average Recipe size in epoch is 10, showing a yellow triangle.:

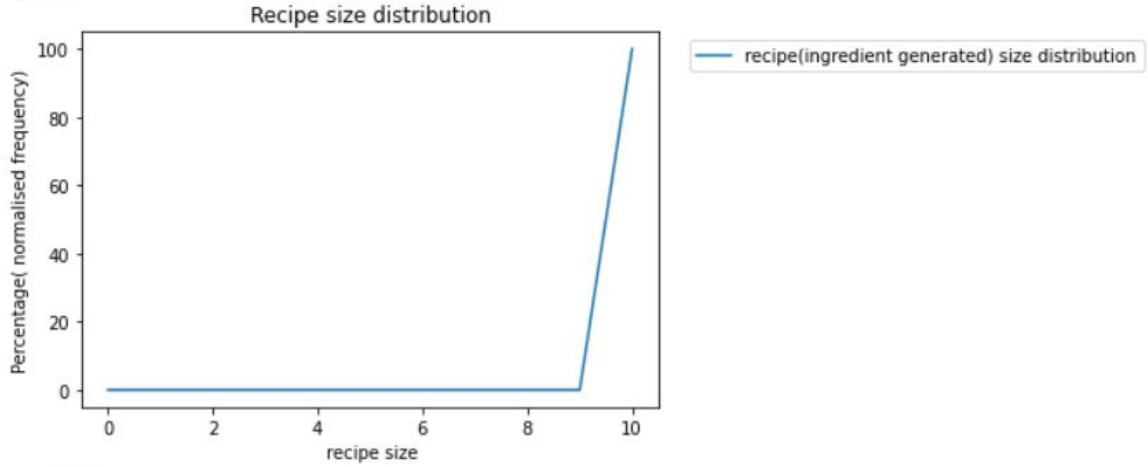
In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.

```
EPOCH:  2
No handles with labels found to put in legend.
```



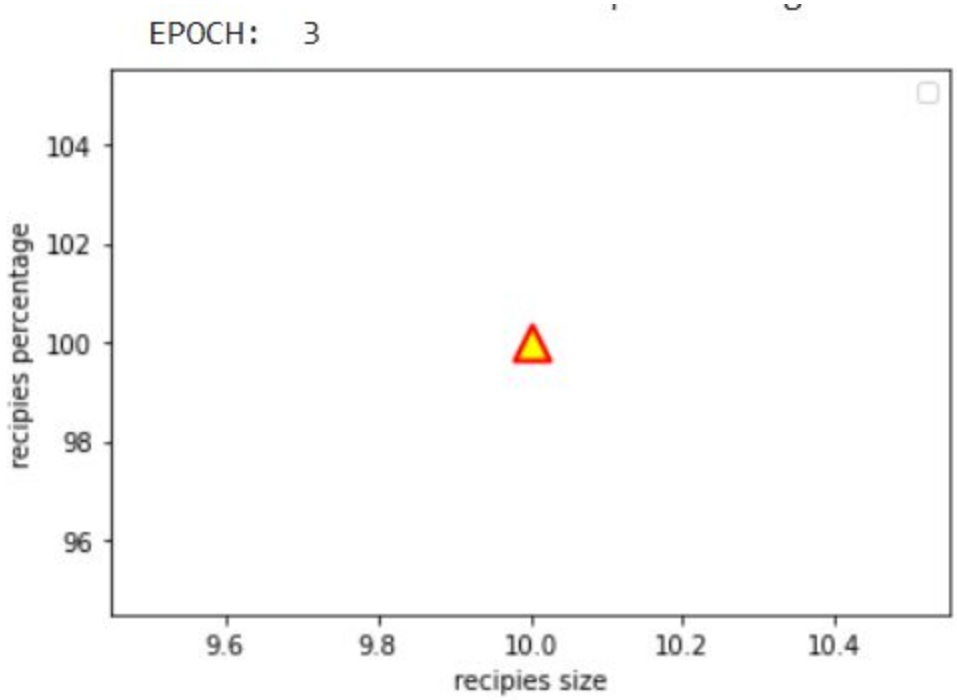
D.Epoch 3 -

```
M ratio:  0.16880374113742647
In epoch:  3
 Number of recipes :  19887
 Size of recipes : 10
 Kitchen bastket size is:  3357
 nature basket :  3357
Epoch:  3
```



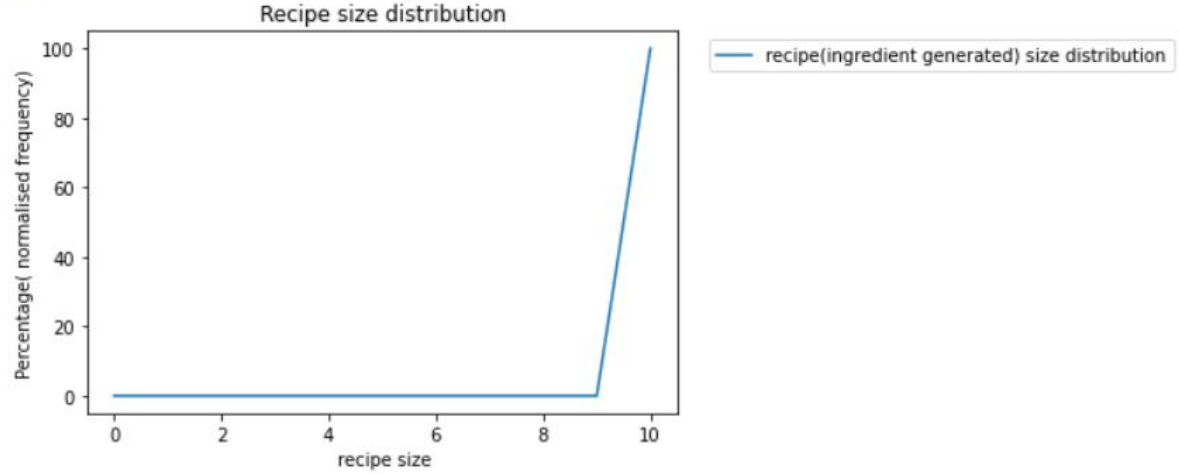

```
X:  4476
 total_number_recipies_generated_so_far:  26516
Nature basket:  2238
 kitchen basket:  4476
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
M ratio:  0.16880374113742647
```

Average recipes size is 10 in recipes list generated so far , showing in yellow triangle.
Analysis:
Average Recipe size in epoch is 10, showing a yellow triangle.:
In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.



E.Epoch 4-

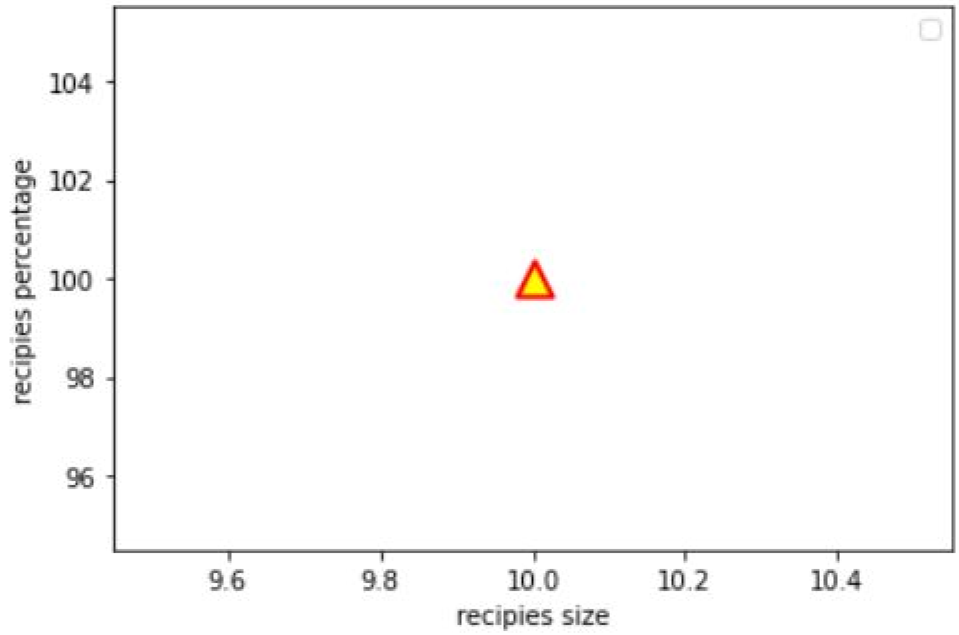
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
M ratio:  0.16880374113742647
In epoch:  4
 Number of recipes :  26516
 Size of recipes : 10
 Kitchen bastket size is:  4476
 nature basket :  2238
Epoch:  4
```




```
X:  5595
 total_number_recipies_generated_so_far:  33145
Nature basket:  1119
 kitchen basket:  5595
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

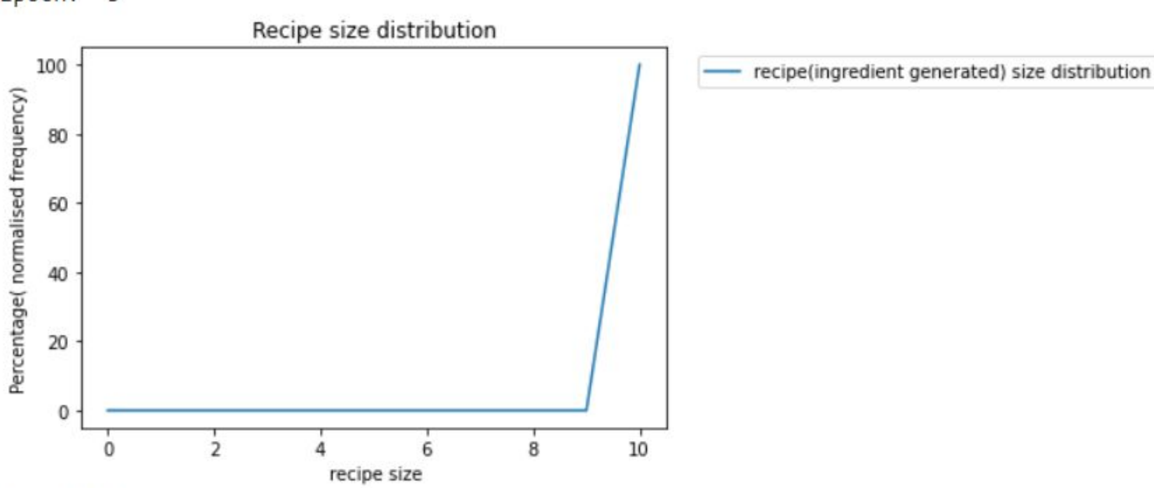
Average recipes size is 10 in recipes list generated so far , showing in yellow triangle.
Analysis:
Average Recipe size in epoch is 10, showing a yellow triangle.:
In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.

```
No handles with labels found to put in legend.
       EPOCH:  4
```



F.Epoch 5-

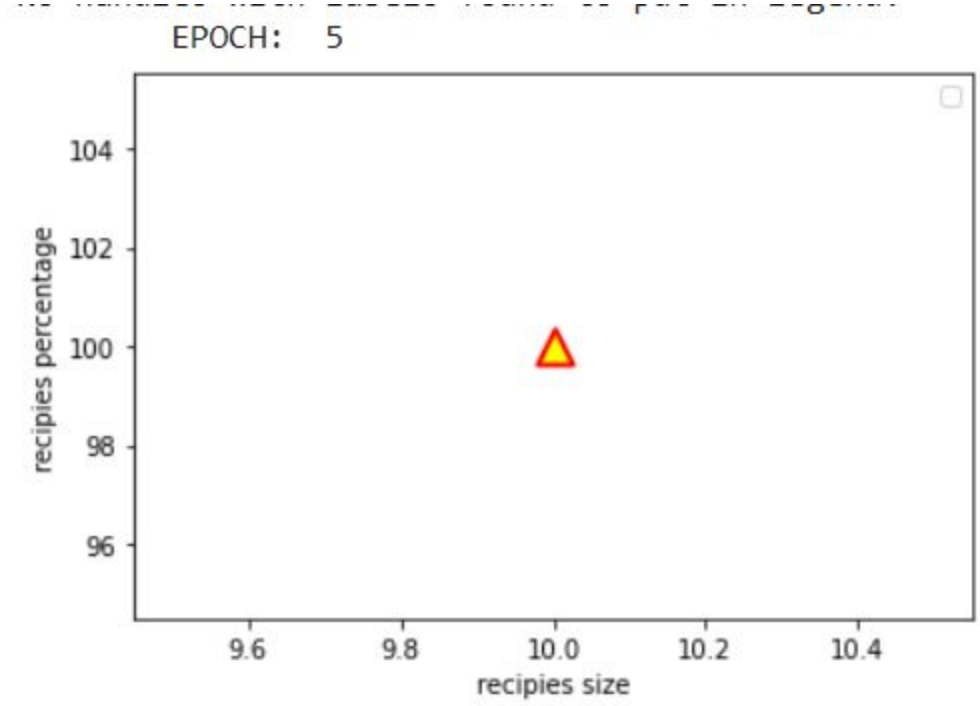
```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
M ratio:  0.16880374113742647
In epoch:  5
 Number of recipes :  33145
 Size of recipes : 10
 Kitchen bastket size is:  5595
 nature basket :  1119
Epoch:  5
```




```
X:  6714
 total_number_recipies_generated_so_far:  39774
Nature basket:  0
 kitchen basket:  6714
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Average recipes size is 10 in recipes list generated so far , showing in yellow triangle.

Analysis:
Average Recipe size in epoch is 10, showing a yellow triangle.:
In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.
Here at the end 39774 new recipes are generated all recipe size is 10.



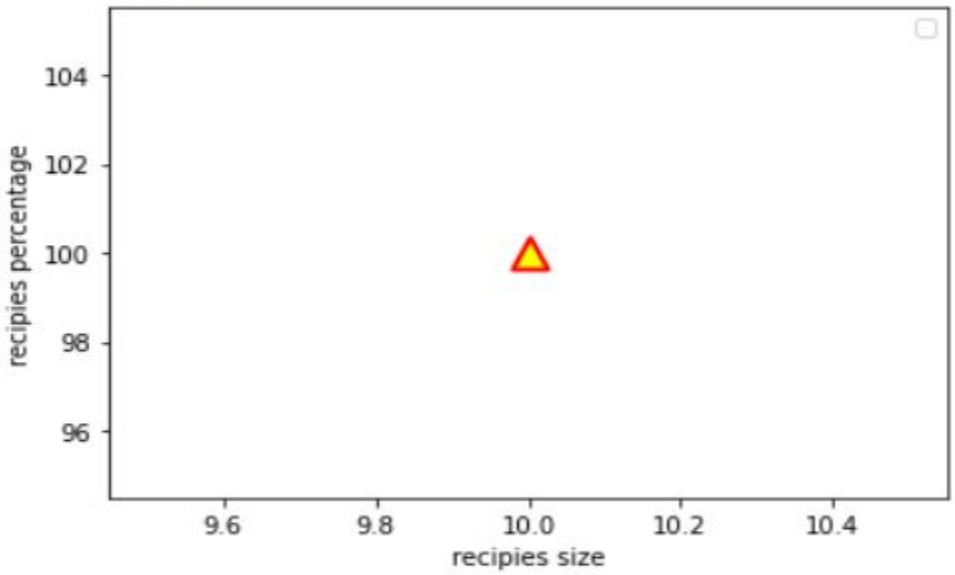
2. Proper Initiation of Variables (Randomised Set of recipes)
A.Epoch:0
Epoch 0 is for initialization of variable,
Size of recipes is= 10
Total number of recipes initialization=500

```
In epoch:  0
 Number of recipes :  500
 Size of recipes : 10
 Kitchen bastket size is:  50
 nature basket :  6664
```

In this step primordial cuisine is my R_0 which is make 500 random different sets of recipes from initials 50 ingredients as shown in code below:

7.R_0=randomly sample S_bar ingredients n times from I_0.

```
[16] R_0=[]
     S_bar=10
     n=500
     for i in range(0,n):
         recip=[]
         while len(recip)!=S_bar:
           index=random.randint(0,len(I_0)-1)
           if I_0[index] in recip:
              continue
           else:
              recip.append(I_0[index])
         R_0.append(set(recip))
```



3. Implementation Copy Mutate of algorithm

It is mentioned in the commented form in CODE uploaded.

Following is code of copy mutation algorithm present in ipynb(please refer):

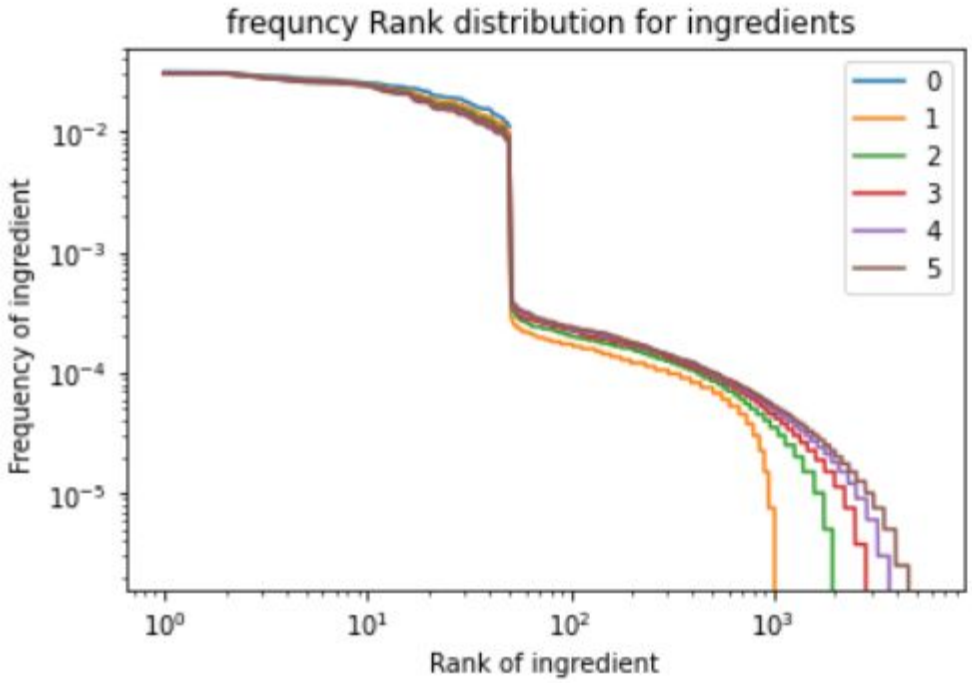
```
for l in range(0, total_epoch):
  count_r=0
  if l!=0:
    count_r=int(len(data_train)/total_epoch)
  else:
    count_r=int(len(data_train)/total_epoch)-n

  print("M ratio: ",(len(I_0)/len(R_0)))
  print("In epoch: ",l)
  print(" Number of recipes : ",len(R_0))
  print(" Size of recipes :",len(R_0[0]))
  print(" Kitchen bastket size is: ",len(I_0))
  print(" nature basket : ",len(I))
  while(len(R_0)!=count_r*(l+1)):
    # delta=len(I_0)/len(R_0)
    # if delta>= phi:
      index=random.randint(0,len(R_0)-1)
      r=list(R_0[index])
      # M=2 here so mutation is 1 time
      while r == list(R_0[index]) :
        index_i=random.randint(0,len(r)-1)
        index_j=random.randint(0,len(I_0)-1)
        i=r[index_i]
        j=I_0[index_j]
        # print("i-",i)
        # print("j-",j)

        if  ( j not in r) and fitness_ingred[j] > fitness_ingred[i] :
          r[index_i]=I_0[index_j]

      if r not in R_0:
        R_0.append(set(r))

    # else:
    # kitchen basket updated:

  R_0_graph.append(R_0)
  I_0_graph.append(I_0)
  recipie_size_plot(R_0,l)
  frquency_rank(R_0,I_0,l)
```

```
  size_NB=6714
  total_number_recipies_dataset=len(data_train)
  total_number_recipies_generated_so_far=len(R_0)
  if l==0:

total_number_recipies_generated_so_far=total_number_recipies_generated_so_
far+500

X=int((size_NB/total_number_recipies_dataset)*total_number_recipies_genera
ted_so_far)
  print("X: ",X)
  print(" total_number_recipies_generated_so_far:
",total_number_recipies_generated_so_far)
  if len(I_0)<X:
    while len(I_0)!=X:
      index_p=random.randint(0,len(I)-1)
      I_0.append(I[index_p])
      I=list(set(I)-{I[index_p]})
```

4. Analyze the output at the end of each Epoch .

-Frequency rank distribution analysis:
In the below graph their is exponential decrementr i.e. steep down fall because ingredient are not uniform in count some have high frequency and some have low frequency , result in exponential decrement.
-Here there is overlap initially of lines because in epoch 2 have all ingredients in epoch 1 , and epoch 3 have all ingredients of epoch 0 ,1,2 and so on at the epoch 5 have all the ingredient of before made cuisines ingredient.
- As epochs get increased the coverage of the x-axis gets increased because as epoch increases recipes of primordial cuisine as well as ingredients are increased.

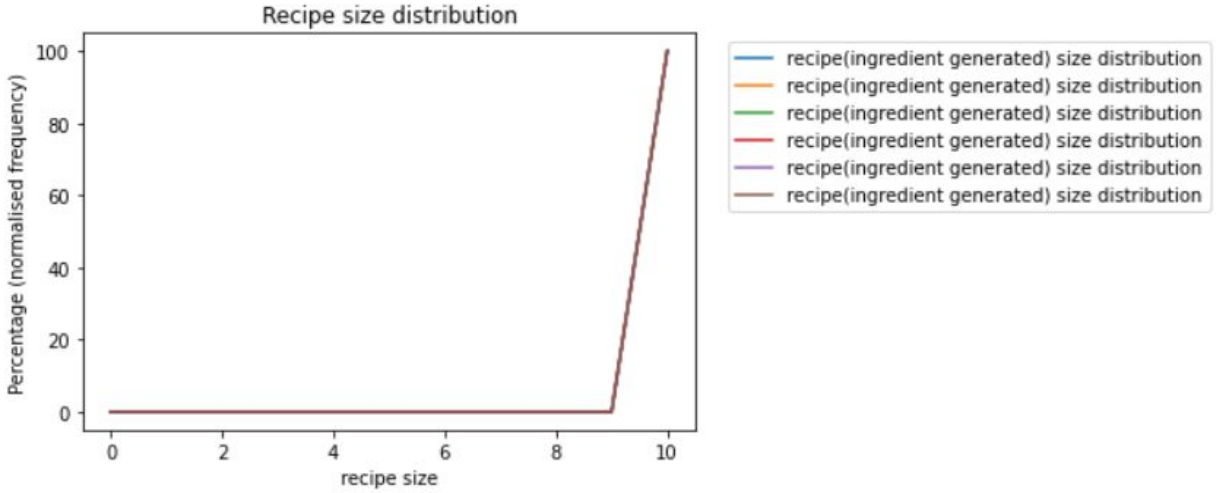
```
import matplotlib.pyplot as plt
for i in range(0,len(A_graph)):
  plt.loglog(A_graph[i], B_graph[i], label =i)
  plt.xlabel('Rank of ingredient')
  plt.ylabel('Frequency of ingredient ')
  plt.title('frequncy Rank distribution for ingredients ')
plt.legend()
plt.show()
```



Graph code for frequency rank distribution:

```
import math
def frquency_rank(R_0,I_0,epoch):

    ingred={k:0 for k in I_0}
    for re in R_0:
        for ingre in re:
                ingred[ingre]+=1
    # figure, axes = plt.subplots(figsize = (8,6))
    ingred={k: v for k, v in sorted(ingred.items(),key=lambda item: item[1],reverse=True)}

    b=list(ingred.values())
    d=0
    for i in b:
        d+=i
    a=[ x for x in range(1,len(b)+1)]
    b1=[y/d for y in b]
    A_graph.append(a)
    B_graph.append(b1)
```

```
for y in recp_y:

  plt.plot(y, label='recipe(ingredient generated) size distribution')
  plt.xlabel('recipe size')
  plt.ylabel('Percentage (normalised frequency)')
  plt.title('Recipe size distribution')

plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plt.show()
```



<u>Analysis of recipe size distribution :</u>

Average Recipe size in epoch is 10, showing a yellow triangle.:

In below graph we can analyse that in every new recipe generation we generated recipes size 10 because we didn't change the recipe size here only replacement happens here till new recipes is not generated and then append it in primordial cuisine at the end.
Here at the end 39774 new recipes are generated all recipe size is 10.
Graph code for recipes size distribution:

```
def recipie_size_plot(recipes,epoch):
  print("Epoch: ",epoch)
  rsize=[]
  for i in recipes:
      rsize.append(len(i))

  y_axis=[]
  for i in range(0,max(rsize)+1):
      y_axis.append(rsize.count(i)*100/len(rsize))
  recp_y.append(y_axis)
  plt.plot(y_axis, label='recipe(ingredient generated) size distribution')
  plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
  plt.xlabel(' recipe size')
  plt.ylabel('Percentage( normalised frequency)')
  plt.title('Recipe size distribution')
  plt.show()
```

5. Maintaining the M ratio
-Initialization of M ratio is 50/500 which is 1:10 considered as per instructed.
- M_ratio= (size of Kitchen basket/ total number of recipes generated so far)
-In Each epoch ratio is about 0.168 approx if I take 6 epochs.(show in above prints)
-M ration is maintained by following formula used in **coding:**

```
size_NB=len(I)
total_number_recipies_dataset=len(data_train)
total_number_recipies_generated_so_far=len(R_0)
X=int((size_NB/total_number_recipies_dataset)*total_number_recipies_genera
ted_so_far)
```

Ans 2-

1. Incorporation of addition, deletion and replacement of ingredients with varying probability:

-In this copy mutation algorithm everything is the same as replacement except here I consider the probability.
-I gave 25% probability to addition , 25% probability to deletion , and 50 % probability to replacement .
-And In Recipes generation loop I had picked up the choices where addition , deletion and replacement had probability as mentioned above , for implementing it I used a list of strings where I put.

```
Assigning probability of:

Take a list of  choices=['addition','deletion','replace','replace'] whose
size is 4. So probability of each calculated as below:

addition : 1/4: 25%

deletion: 1/4 : 25%

replace : 2/4 : 50%
```

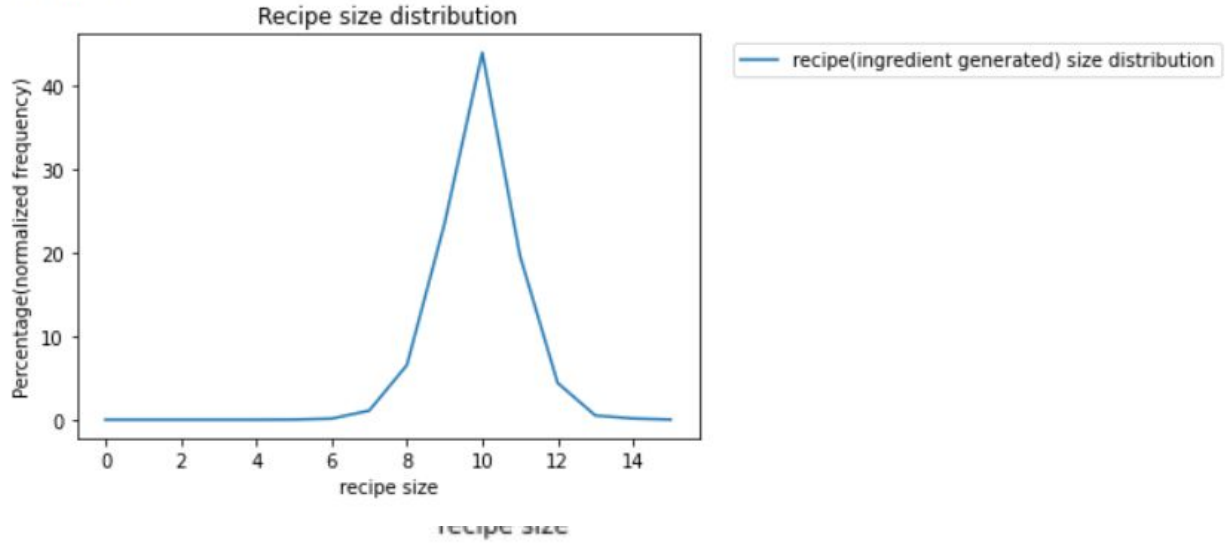
-And we picked each choice with the above probabilities.
-As mentioned in the below code :
We choose our operations choices (replace, addition and deletion) in each recipe updation loop and inside that we replaced or added in any randomly chosen probability until new recipe is not generated , and after once we get new recipe we updated it in our primordial cuisine set i.e. update recipes and at the and we get 39774 new recipes and using all ingredients.

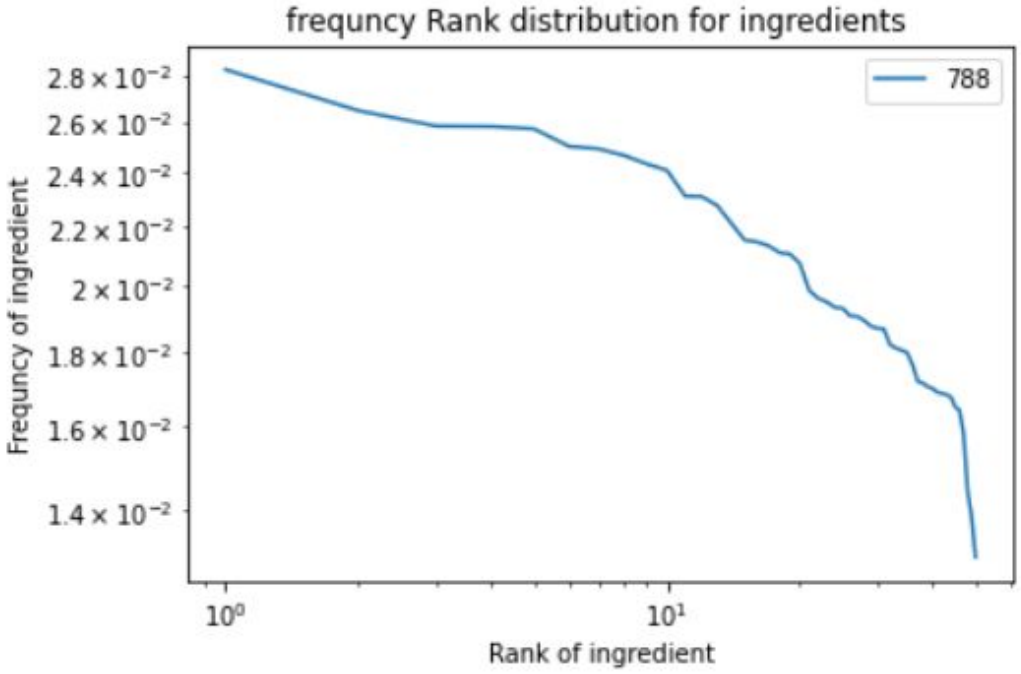
```
while(len(R_0)!=count_r*(l+1)):
  # delta=len(I_0)/len(R_0)
  # if delta>= phi:
    index_choice=random.randint(0,len(choices)-1)
    index=random.randint(0,len(R_0)-1)
    r=list(R_0[index])
    if len(r)<2:
      continue
    # # M=2 here so mutation is 1 time
    # if choices[index_choice]=='addition' or choices[index_choice]=='deletion':
    #   v=1
    # else:
    #   v=10
    # # M=2 here so mutation is 1 time
    while r == list(R_0[index]) :
      index_i=random.randint(0,len(r)-1)
      index_j=random.randint(0,len(I_0)-1)
      i=r[index_i]
      j=I_0[index_j]
      # print("i-",i)
      # print("j-",j)

      if choices[index_choice]=='replace':
        if  ( j not in r) and fitness_ingred[j] > fitness_ingred[i] :
          r[index_i]=I_0[index_j]
      if choices[index_choice]=='addition':
          r.append(I_0[index_j])
      if choices[index_choice]=='deletion':
          r.remove(r[index_i])
```

2. **Cuisine analysis per epoch** (depends on the number of epochs mentioned) recipe size distribution and frequency- rank statistics at the end of each Epoch.)

A.Epoch 0:

```
epoch:  0
RATIO OF M:  0.1
 Number of recipes :  500
 Size of recipes : 10
 Kitchen bastket size is:  50
 nature basket :  6664
Epoch:  0
```

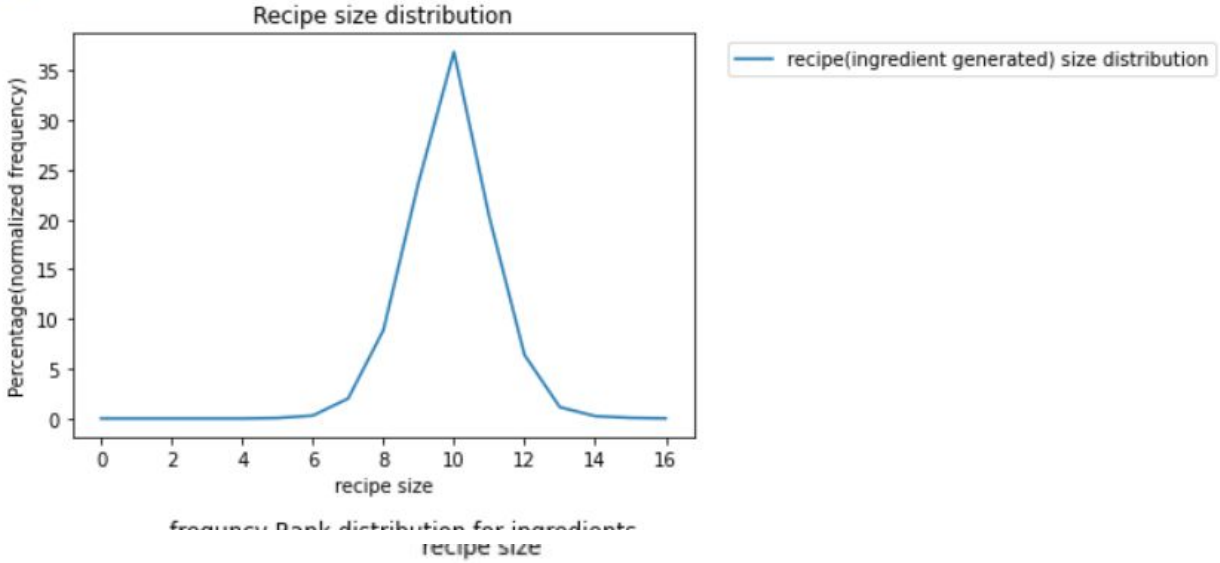


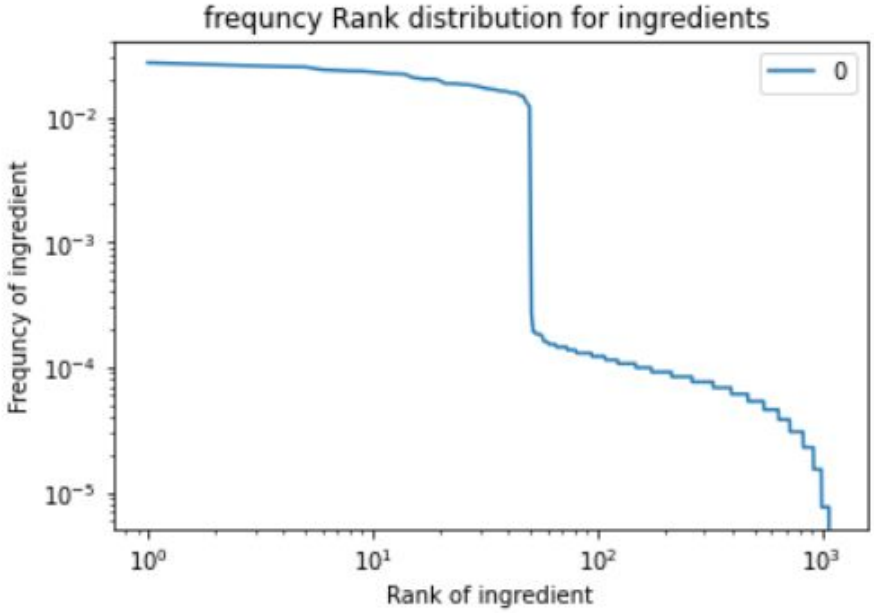



```
X:  1119
 total_number_recipies_generated_so_far:  6629
Nature basket:  5595
 kitchen basket:  1119
```

B.Epoch 1:

```
epoch:  1
RATIO OF M:  0.18257464512971122
 Number of recipes :  6129
 Size of recipes : 10
 Kitchen bastket size is:  1119
 nature basket :  5595
Epoch:  1
```

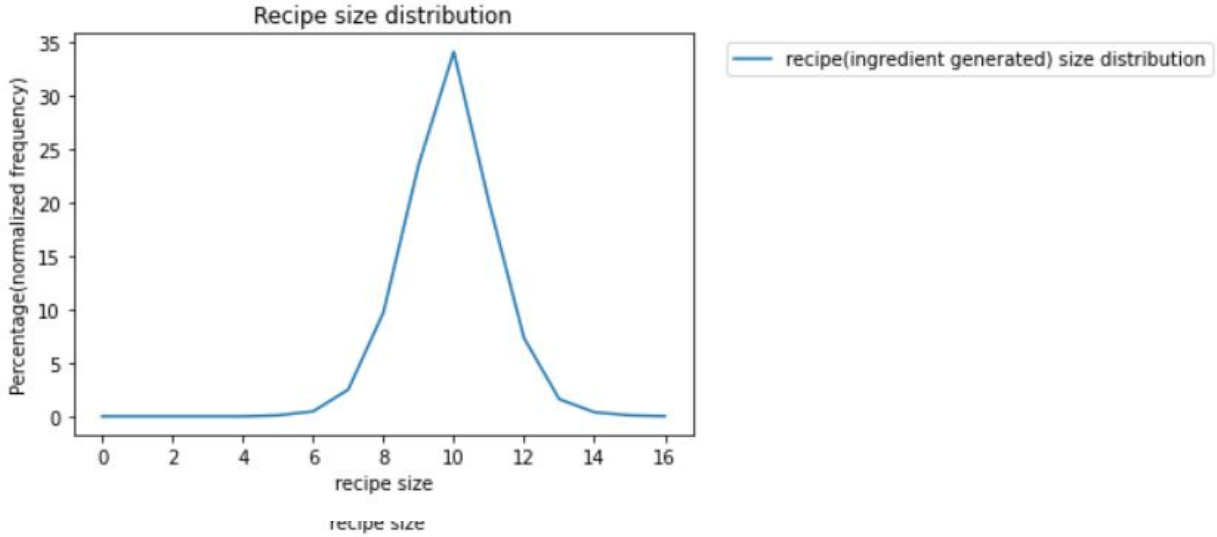


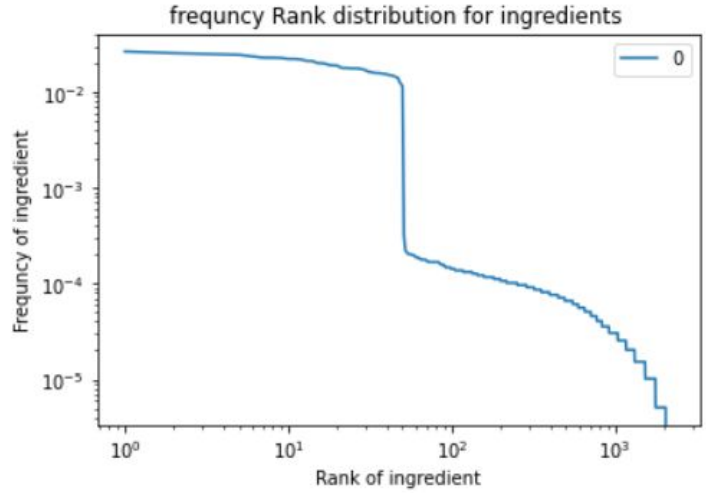



```
X:  2238
 total_number_recipies_generated_so_far:  13258
Nature basket:  4476
 kitchen basket:  2238
```

C.Epoch 2:

```
epoch:  2
RATIO OF M:  0.16880374113742647
 Number of recipes :  13258
 Size of recipes : 10
 Kitchen bastket size is:  2238
 nature basket :  4476
Epoch:  2
```

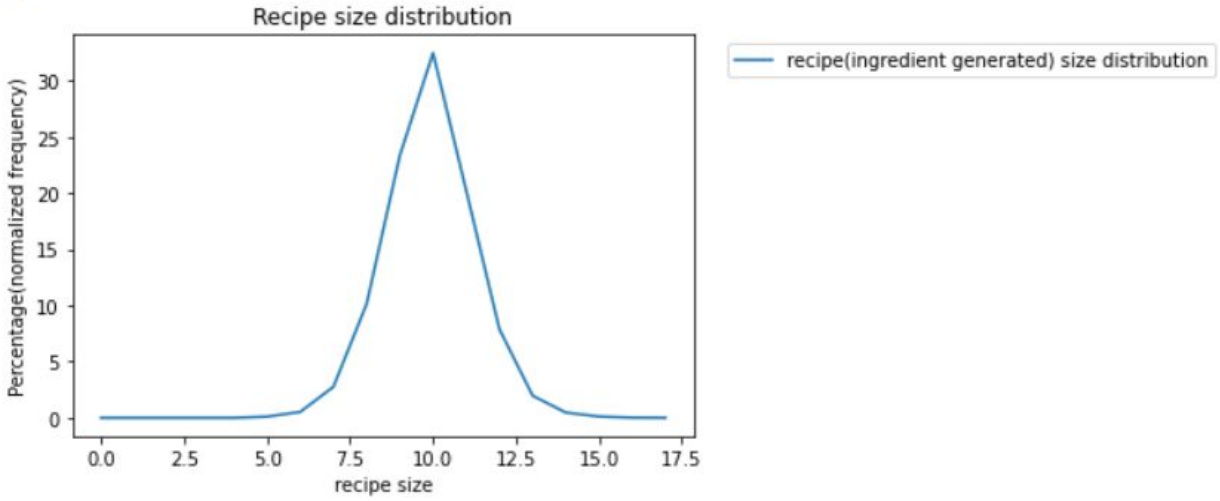


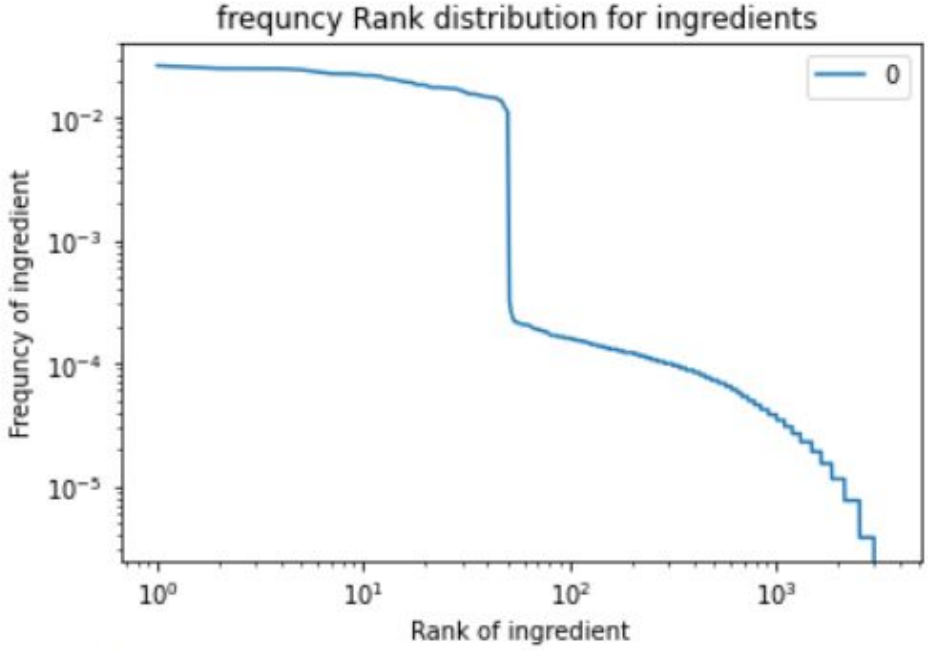



```
X:  3357
 total_number_recipies_generated_so_far:  19887
Nature basket:  3357
 kitchen basket:  3357
```

D.Epoch 3:

```
epoch:  3
RATIO OF M:  0.16880374113742647
 Number of recipes :  19887
 Size of recipes : 10
 Kitchen bastket size is:  3357
 nature basket :  3357
Epoch:  3
```

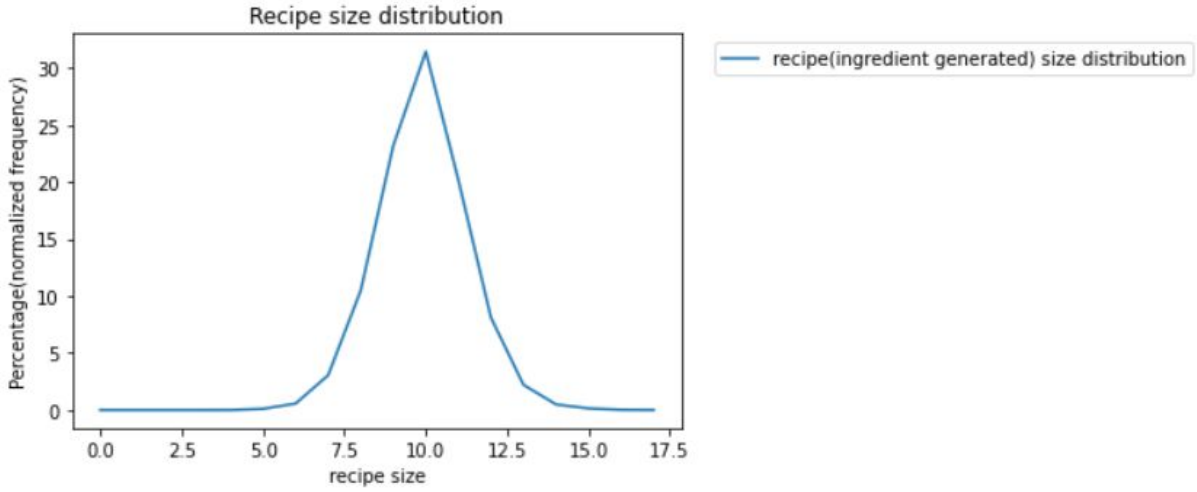


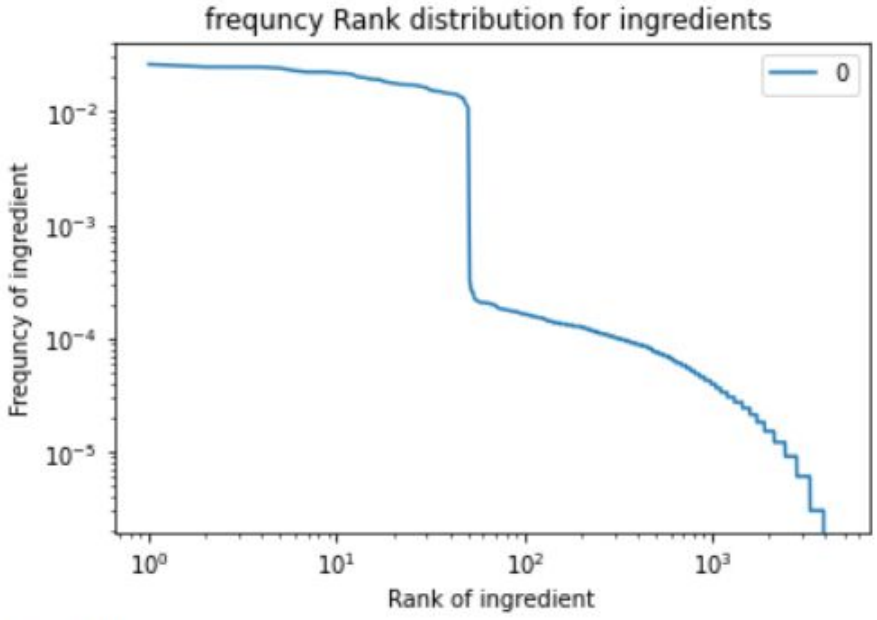



```
X:  4476
 total_number_recipies_generated_so_far:  26516
Nature basket:  2238
 kitchen basket:  4476
```

E.Epoch 4:

```
epoch:  4
RATIO OF M:  0.16880374113742647
 Number of recipes :  26516
 Size of recipes : 10
 Kitchen bastket size is:  4476
 nature basket :  2238
Epoch:  4
```

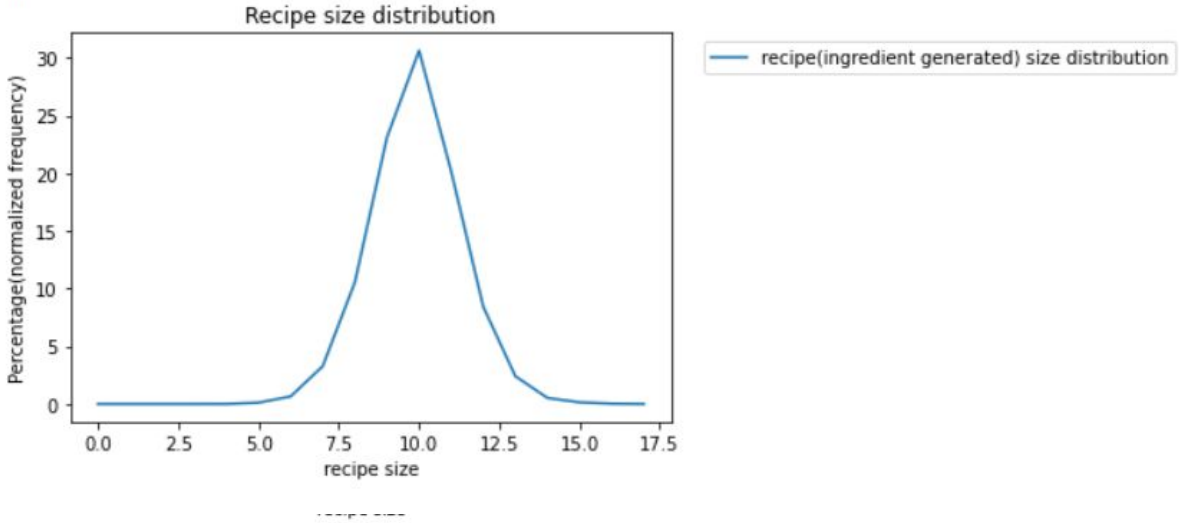


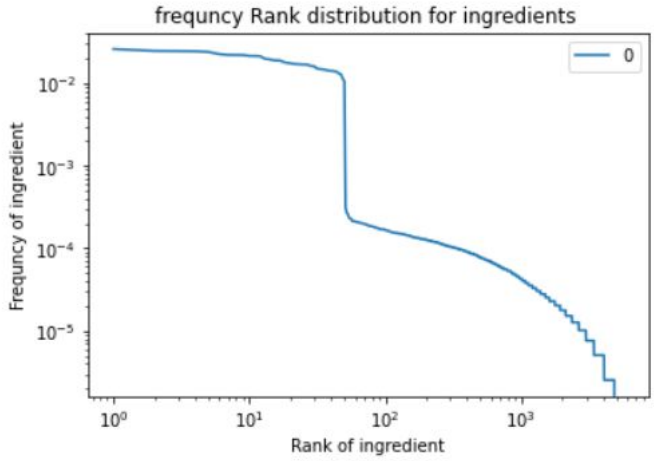



```
X:  5595
 total_number_recipies_generated_so_far:  33145
Nature basket:  1119
 kitchen basket:  5595
```

F.Epoch 5:

```
epoch:  5
RATIO OF M:  0.16880374113742647
 Number of recipes :  33145
 Size of recipes : 10
 Kitchen bastket size is:  5595
 nature basket :  1119
Epoch:  5
```






```
X:  6714
 total_number_recipies_generated_so_far:  39774
Nature basket:  0
 kitchen basket:  6714
```

<u>-Frequency Rank distribution for each epoch:</u>

-Frequency rank distribution analysis:

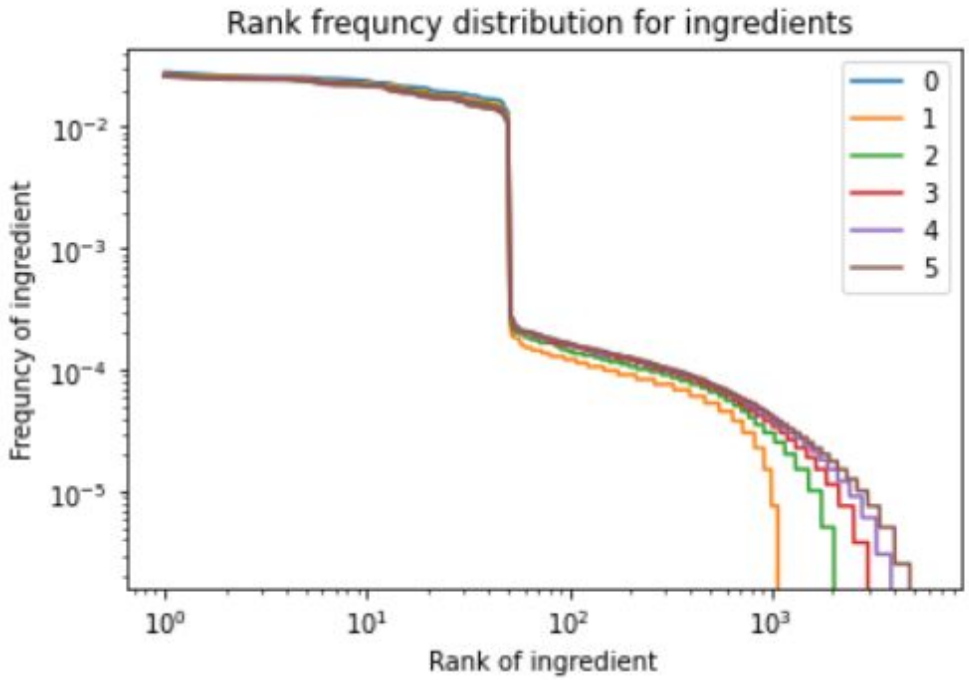
In the below graph their is exponential decrementr i.e. steep down fall because ingredient are not uniform in count some have high frequency and some have low frequency , result in exponential decrement.

-Here there is overlap initially of lines because in epoch 2 have all ingredients in epoch 1 , and epoch 3 have all ingredients of epoch 0 ,1,2 and so on at the epoch 5 have all the ingredient of before made cuisines ingredient.

- As epochs get increased the coverage of the x-axis gets increased because as epoch increases recipes of primordial cuisine as well as ingredients are increased.

Legends shown in graph 0 to 5 are epochs from 0 to 5.

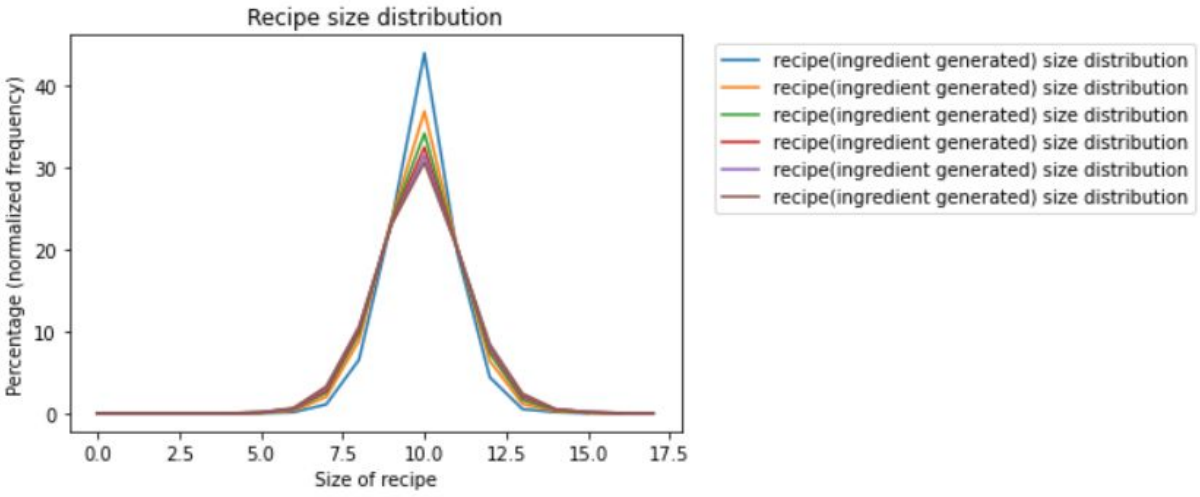
```
import matplotlib.pyplot as plt
for i in range(0,len(A_graph)):
  plt.loglog(A_graph[i], B_graph[i], label =i)
  plt.xlabel('Rank of ingredient')
  plt.ylabel('Frequncy of ingredient')
  plt.title('Rank frequncy distribution for ingredients ')
plt.legend()
plt.show()
```



-Recipies size distribution for each epoch
Analysis :

-In this size of recipes vary from 1 to any greater size because here we are doing 3 operation addition , deletion or replacement of ingredients with some probability of 25%,25% or 50%.
- In the initial epoch most of the time recipes size is 10 because initially recipes size is 10 and on further epoch deletion and addition are increased so recipes sizes are decreased in further bessie deletion and addition are increased.
-Here most of the time the size of recipes is 10 because we are given 50% probability of choice for replacement , and 25%,25% for deletion and addition.

```
for y in recp_y:

  plt.plot(y, label='recipe(ingredient generated) size distribution')
  plt.xlabel('Size of recipe')
  plt.ylabel('Percentage (normalized frequency)')
  plt.title('Recipe size distribution')

plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
plt.show()
```



3.Time Complexity Analysis:
-Time complexity is the number of recipes in the dataset.
-Time complexity :
No. of epochs are =Epoch
Number of recipes in train data=N
Then: number of recipes to be generated in each epoch are=Epoch/N

-kitchen basket size=K_B
-nature basket size(overall ingredient in train data)=N_B

Then:
case:Time complexity for best case is-
=N*(N/Epoch)*Epoch
=N square

case:Time complexity for worst case is-
=no. Of epoch *(total recipes generated in each epoch + N_B)*N
-There changes may occur because of fitness value comparison and fitness value might be different then the existing one.
-in worst case also we check all the ingredients' nature as replacement because addition and deletion didn't take extra time, so nature is the same in all operations.