# 4 Task 4

## 4.1 a)

```
#scale
cows["age_scaler"]=(cows["age"]-cows["age"].min())/(cows["age"].max()-cows["age"].min())
cows["milk_scaler"]=(cows["milk"]-cows["milk"].min())/(cows["milk"].max()-cows["milk"].min())
cows

ndata=np.array(cows[["age_scaler","milk_scaler"]])

# Compute numerical similarity
def get_numsim_M(ndata):
    lens=len(ndata)

    SM=np.zeros((lens,lens))

    for i in range(lens):
        j=i+1
        while j <lens:
            tep=(ndata[i][0]-ndata[j][0])**2+(ndata[i][1]-ndata[j][1])**2
            tep=np.sqrt(tep)
            SM[i][j]=SM[j][i]=tep
            j=j+1
    return SM.round(4)

re=get_numsim_M(ndata)
re
```

The euclidean distances between cows is shown in Fig.4

```
array([[0.    , 0.2857, 0.52  , 0.3627, 0.9371, 1.2289],
       [0.2857, 0.    , 0.52  , 0.5429, 1.0976, 1.4142],
       [0.52  , 0.52  , 0.    , 0.3308, 0.6615, 0.9923],
       [0.3627, 0.5429, 0.3308, 0.    , 0.5759, 0.8781],
       [0.9371, 1.0976, 0.6615, 0.5759, 0.    , 0.3308],
       [1.2289, 1.4142, 0.9923, 0.8781, 0.3308, 0.    ]])
```

Figure 4: Euclidean distance

## 4.2 b)

The Goodall distances between cows is shown in Fig.5

```
array([[0.    , 0.7037, 0.7037, 1.    , 1.    , 1.    ],
       [0.7037, 0.    , 1.    , 0.75  , 1.    , 0.4537],
       [0.7037, 1.    , 0.    , 0.75  , 0.75  , 0.7037],
       [1.    , 0.75  , 0.75  , 0.    , 0.4537, 0.75  ],
       [1.    , 1.    , 0.75  , 0.4537, 0.    , 1.    ],
       [1.    , 0.4537, 0.7037, 0.75  , 1.    , 0.    ]])
```

Figure 5: Goodall distances

```
#construct dict
dic={"Ayrshir": 1/2,"Holstein": 1/3,"Finncattle": 1/6,
        "lively":1/6,"kind":1/3,"calm":1/2,
        "rock":1/3,"country":1/3,"classical":1/3}

#Get categorical features
ccols=["race","character","music"]
cdata=np.array(cows[ccols])
cdata

#Compute categorical similarity
def get_catsim_M(cdata,dic):
    lens,f_num=cdata.shape

    CM=np.zeros((lens,lens))

    for i in range(lens):
        j=i+1
        while j <lens:
            tep=0
            for x in range(f_num):
                if cdata[i][x]==cdata[j][x]:
                    tep=tep+1-dic[cdata[i][x]]**2
            CM[i][j]=CM[j][i]=1-tep/f_num
            j=j+1
    return CM.round(4)

re=get_catsim_M(cdata,dic)

re
```

## 4.3 c)

Distance using both numerical and categorical features is shown in Fif.6

```
array([[0.   , 1.507, 1.735, 2.1  , 2.658, 2.941],
       [1.507, 0.   , 2.253, 1.838, 2.814, 2.167],
       [1.735, 2.253, 0.   , 1.632, 1.953, 2.194],
       [2.1  , 1.838, 1.632, 0.   , 1.352, 2.164],
       [2.658, 2.814, 1.953, 1.352, 0.   , 2.069],
       [2.941, 2.167, 2.194, 2.164, 2.069, 0.   ]])
```

Figure 6: Combined distance measure

```
numsim=get_numsim_M(ndata)
catsim=get_catsim_M(cdata,dic)

a=2/5

mixSim=a*numsim/np.std(numsim)+(1-a)*catsim/np.std(catsim)
mixSim=mixSim.round(3)
mixSim
```

## 4.4 d)

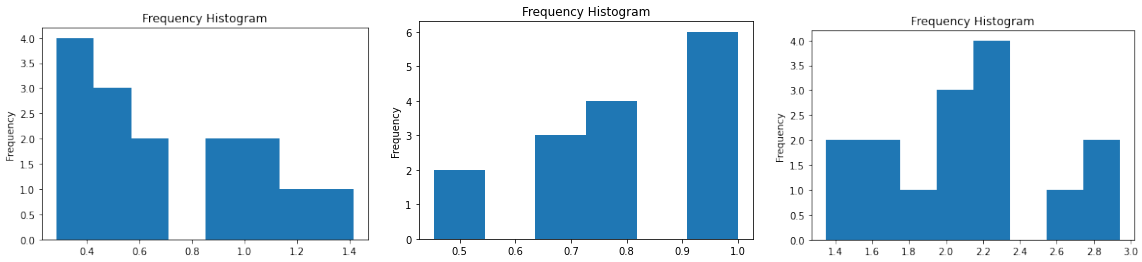
### 4.4.1 Histograms

```
#Show Histograms
def show_hist(data,bins):
    x =[]
    for i in range(6):
        x.extend(data[i][i+1:6])
    # print(x)
    plt.hist(x, bins=bins)
    plt.gca().set(title='Frequency Histogram', ylabel='
       Frequency')
    plt.show()
```

After test different numbers of bins in all three cases, I choose 8,6,8 in each plot as their number of bins to illustrate the clusters of cows in data. The figures are shown in Fig.7 I think the measurement combined numerical and categorical features can better cluster cows.Because more features are used for calculation, it means that we have considered more comprehensively, resulting better effect.

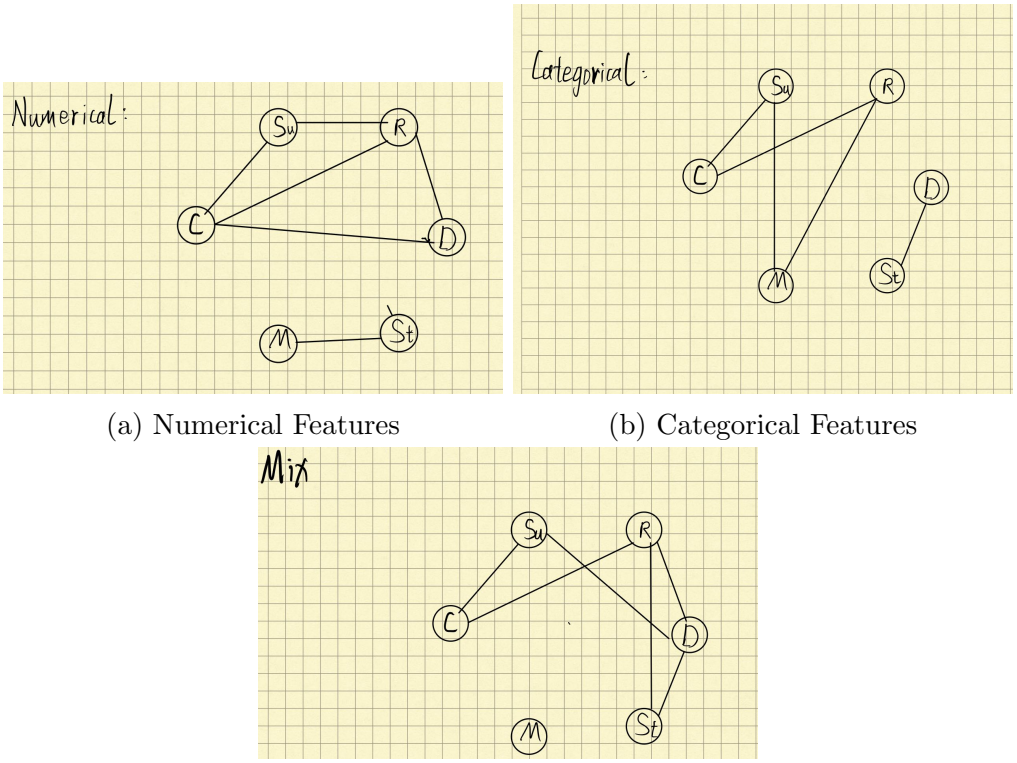
### 4.4.2 Graph-based clustering

First we can create a complete distance graph between cows. and we know the distances of cows from the task above. Then We should remove edges with longest distances(largest




(a) Numerical Features, Bins:8 (b) Categorical Features, Bins:6 (c) Numerical+Categorical Features, Bins:8

Figure 7: Histograms in 3 different cases




(a) Numerical Features (b) Categorical Features

(c) Numerical+Categorical Features

Figure 8: Graph clustering in 3 different cases

weights) until the graph is broken into two connected components. These components are clusters of cows.I think the measurement of mix features including numerical and categorical can better cluster cows from Fig.8.The results are not so radical to make the number of cows out of balance among the different clusters. It balances the application of two different types of data into the similarity, the more data types are used in the results, making the results more inclusive.