# Healthcare

August 28, 2022

### 0.0.1 Input the required libraries

[1]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
%matplotlib inline

from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
vectorizer = np.vectorize(lambda x: colors[x % len(colors)])
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn import ensemble
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.metrics import  classification_report

import warnings
warnings.filterwarnings(action='ignore',category=DeprecationWarning)
warnings.filterwarnings(action='ignore',category=FutureWarning)


import warnings
warnings.filterwarnings('ignore')
```

### Load Data and labels

[2]:
```
label = pd.read_csv('labels.csv')
data = pd.read_csv('data.csv')
```

[3]:
```
data.describe()
```

[3]:

| | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | \ |
|---|---|---|---|---|---|---|---|
| count | 801.000000 | 801.000000 | 801.000000 | 801.000000 | 801.000000 | 801.0 | |
| mean | 0.026642 | 3.010909 | 3.095350 | 6.722305 | 9.813612 | 0.0 | |
| std | 0.136850 | 1.200828 | 1.065601 | 0.638819 | 0.506537 | 0.0 | |
| min | 0.000000 | 0.000000 | 0.000000 | 5.009284 | 8.435999 | 0.0 | |
| 25% | 0.000000 | 2.299039 | 2.390365 | 6.303346 | 9.464466 | 0.0 | |
| 50% | 0.000000 | 3.143687 | 3.127006 | 6.655893 | 9.791599 | 0.0 | |
| 75% | 0.000000 | 3.883484 | 3.802534 | 7.038447 | 10.142324 | 0.0 | |
| max | 1.482332 | 6.237034 | 6.063484 | 10.129528 | 11.355621 | 0.0 | |

| | gene_6 | gene_7 | gene_8 | gene_9 | … | gene_20521 | \ |
|---|---|---|---|---|---|---|---|
| count | 801.000000 | 801.000000 | 801.000000 | 801.000000 | … | 801.000000 | |
| mean | 7.405509 | 0.499882 | 0.016744 | 0.013428 | … | 5.896573 | |
| std | 1.108237 | 0.508799 | 0.133635 | 0.204722 | … | 0.746399 | |
| min | 3.930747 | 0.000000 | 0.000000 | 0.000000 | … | 2.853517 | |
| 25% | 6.676042 | 0.000000 | 0.000000 | 0.000000 | … | 5.454926 | |
| 50% | 7.450114 | 0.443076 | 0.000000 | 0.000000 | … | 5.972582 | |
| 75% | 8.121984 | 0.789354 | 0.000000 | 0.000000 | … | 6.411292 | |
| max | 10.718190 | 2.779008 | 1.785592 | 4.067604 | … | 7.771054 | |

| | gene_20522 | gene_20523 | gene_20524 | gene_20525 | gene_20526 | gene_20527 | \ |
|---|---|---|---|---|---|---|---|
| count | 801.000000 | 801.000000 | 801.000000 | 801.000000 | 801.000000 | 801.000000 | |
| mean | 8.765891 | 10.056252 | 4.847727 | 9.741987 | 11.742228 | 10.155271 | |
| std | 0.603176 | 0.379278 | 2.382728 | 0.533898 | 0.670371 | 0.580569 | |
| min | 6.678368 | 8.669456 | 0.000000 | 7.974942 | 9.045255 | 7.530141 | |
| 25% | 8.383834 | 9.826027 | 3.130750 | 9.400747 | 11.315857 | 9.836525 | |
| 50% | 8.784144 | 10.066385 | 5.444935 | 9.784524 | 11.749802 | 10.191207 | |
| 75% | 9.147136 | 10.299025 | 6.637412 | 10.082269 | 12.177852 | 10.578561 | |
| max | 11.105431 | 11.318243 | 9.207495 | 11.811632 | 13.715361 | 11.675653 | |

| | gene_20528 | gene_20529 | gene_20530 |
|---|---|---|---|
| count | 801.000000 | 801.000000 | 801.000000 |
| mean | 9.590726 | 5.528177 | 0.095411 |
| std | 0.563849 | 2.073859 | 0.364529 |
| min | 7.864533 | 0.593975 | 0.000000 |
| 25% | 9.244219 | 4.092385 | 0.000000 |
| 50% | 9.566511 | 5.218618 | 0.000000 |
| 75% | 9.917888 | 6.876382 | 0.000000 |
| max | 12.813320 | 11.205836 | 5.254133 |

[8 rows x 20531 columns]

### 0.0.2 Exploratory Data Analysis:

### 0.0.3 1.Merge both the datasets.

[4]:
```
merged_data = pd.merge(label,data)
```

[5]:
```
merged_data.head()
```

[5]:
```
  Unnamed: 0 Class  gene_0    gene_1    gene_2    gene_3     gene_4  gene_5  \
0   sample_0  PRAD     0.0  2.017209  3.265527  5.478487  10.431999     0.0
1   sample_1  LUAD     0.0  0.592732  1.588421  7.586157   9.623011     0.0
2   sample_2  PRAD     0.0  3.511759  4.327199  6.881787   9.870730     0.0
3   sample_3  PRAD     0.0  3.663618  4.507649  6.659068  10.196184     0.0
4   sample_4  BRCA     0.0  2.655741  2.821547  6.539454   9.738265     0.0

     gene_6    gene_7  …  gene_20521  gene_20522  gene_20523  gene_20524  \
0  7.175175  0.591871  …    4.926711    8.210257    9.723516    7.220030
1  6.816049  0.000000  …    4.593372    7.323865    9.740931    6.256586
2  6.972130  0.452595  …    5.125213    8.127123   10.908640    5.401607
3  7.843375  0.434882  …    6.076566    8.792959   10.141520    8.942805
4  6.566967  0.360982  …    5.996032    8.891425   10.373790    7.181162

   gene_20525  gene_20526  gene_20527  gene_20528  gene_20529  gene_20530
0    9.119813   12.003135    9.650743    8.921326    5.286759         0.0
1    8.381612   12.674552   10.517059    9.397854    2.094168         0.0
2    9.911597    9.045255    9.788359   10.090470    1.683023         0.0
3    9.601208   11.392682    9.694814    9.684365    3.292001         0.0
4    9.846910   11.922439    9.217749    9.461191    5.110372         0.0

[5 rows x 20533 columns]
```

[6]:
```
merged_data.isnull().sum()
```

[6]:
```
Unnamed: 0    0
Class         0
gene_0        0
gene_1        0
gene_2        0
             ..
gene_20526    0
gene_20527    0
gene_20528    0
gene_20529    0
gene_20530    0
Length: 20533, dtype: int64
```

[7]:
```
merged_data.describe()
```

[7]:
```
            gene_0      gene_1      gene_2      gene_3      gene_4  gene_5  \
count  801.000000  801.000000  801.000000  801.000000  801.000000   801.0
mean     0.026642    3.010909    3.095350    6.722305    9.813612     0.0
std      0.136850    1.200828    1.065601    0.638819    0.506537     0.0
min      0.000000    0.000000    0.000000    5.009284    8.435999     0.0
25%      0.000000    2.299039    2.390365    6.303346    9.464466     0.0
50%      0.000000    3.143687    3.127006    6.655893    9.791599     0.0
75%      0.000000    3.883484    3.802534    7.038447   10.142324     0.0
max      1.482332    6.237034    6.063484   10.129528   11.355621     0.0

            gene_6      gene_7      gene_8      gene_9  …  gene_20521  \
count  801.000000  801.000000  801.000000  801.000000  …  801.000000
mean     7.405509    0.499882    0.016744    0.013428  …    5.896573
std      1.108237    0.508799    0.133635    0.204722  …    0.746399
min      3.930747    0.000000    0.000000    0.000000  …    2.853517
25%      6.676042    0.000000    0.000000    0.000000  …    5.454926
50%      7.450114    0.443076    0.000000    0.000000  …    5.972582
75%      8.121984    0.789354    0.000000    0.000000  …    6.411292
max     10.718190    2.779008    1.785592    4.067604  …    7.771054

       gene_20522  gene_20523  gene_20524  gene_20525  gene_20526  gene_20527  \
count  801.000000  801.000000  801.000000  801.000000  801.000000  801.000000
mean     8.765891   10.056252    4.847727    9.741987   11.742228   10.155271
std      0.603176    0.379278    2.382728    0.533898    0.670371    0.580569
min      6.678368    8.669456    0.000000    7.974942    9.045255    7.530141
25%      8.383834    9.826027    3.130750    9.400747   11.315857    9.836525
50%      8.784144   10.066385    5.444935    9.784524   11.749802   10.191207
75%      9.147136   10.299025    6.637412   10.082269   12.177852   10.578561
max     11.105431   11.318243    9.207495   11.811632   13.715361   11.675653

       gene_20528  gene_20529  gene_20530
count  801.000000  801.000000  801.000000
mean     9.590726    5.528177    0.095411
std      0.563849    2.073859    0.364529
min      7.864533    0.593975    0.000000
25%      9.244219    4.092385    0.000000
50%      9.566511    5.218618    0.000000
75%      9.917888    6.876382    0.000000
max     12.813320   11.205836    5.254133

[8 rows x 20531 columns]
```

[8]:
```
merged_data.columns
```

[8]:
```
Index(['Unnamed: 0', 'Class', 'gene_0', 'gene_1', 'gene_2', 'gene_3', 'gene_4',
       'gene_5', 'gene_6', 'gene_7',
       …
```

```
       'gene_20521', 'gene_20522', 'gene_20523', 'gene_20524', 'gene_20525',
       'gene_20526', 'gene_20527', 'gene_20528', 'gene_20529', 'gene_20530'],
      dtype='object', length=20533)
```

### 0.0.4 2.Plot the merged dataset as a hierarchically-clustered heatmap.

[9]:
```
heatmap_data = pd.pivot_table(merged_data, index=['Class'])
```

[10]:
```
heatmap_data.head()
```

[10]:

| Class | gene_0 | gene_1 | gene_10 | gene_100 | gene_1000 | gene_10000 | \ |
|---|---|---|---|---|---|---|---|
| BRCA | 0.011362 | 2.839739 | 0.544066 | 10.681488 | 10.303568 | 3.258028 | |
| COAD | 0.022212 | 3.438381 | 0.357278 | 11.015745 | 9.951124 | 3.462039 | |
| KIRC | 0.046544 | 2.398129 | 1.166824 | 10.238999 | 11.148094 | 1.651798 | |
| LUAD | 0.041088 | 3.358260 | 0.607541 | 10.517670 | 10.503698 | 3.754181 | |
| PRAD | 0.026544 | 3.441041 | 0.765608 | 10.282936 | 9.967433 | 1.949878 | |

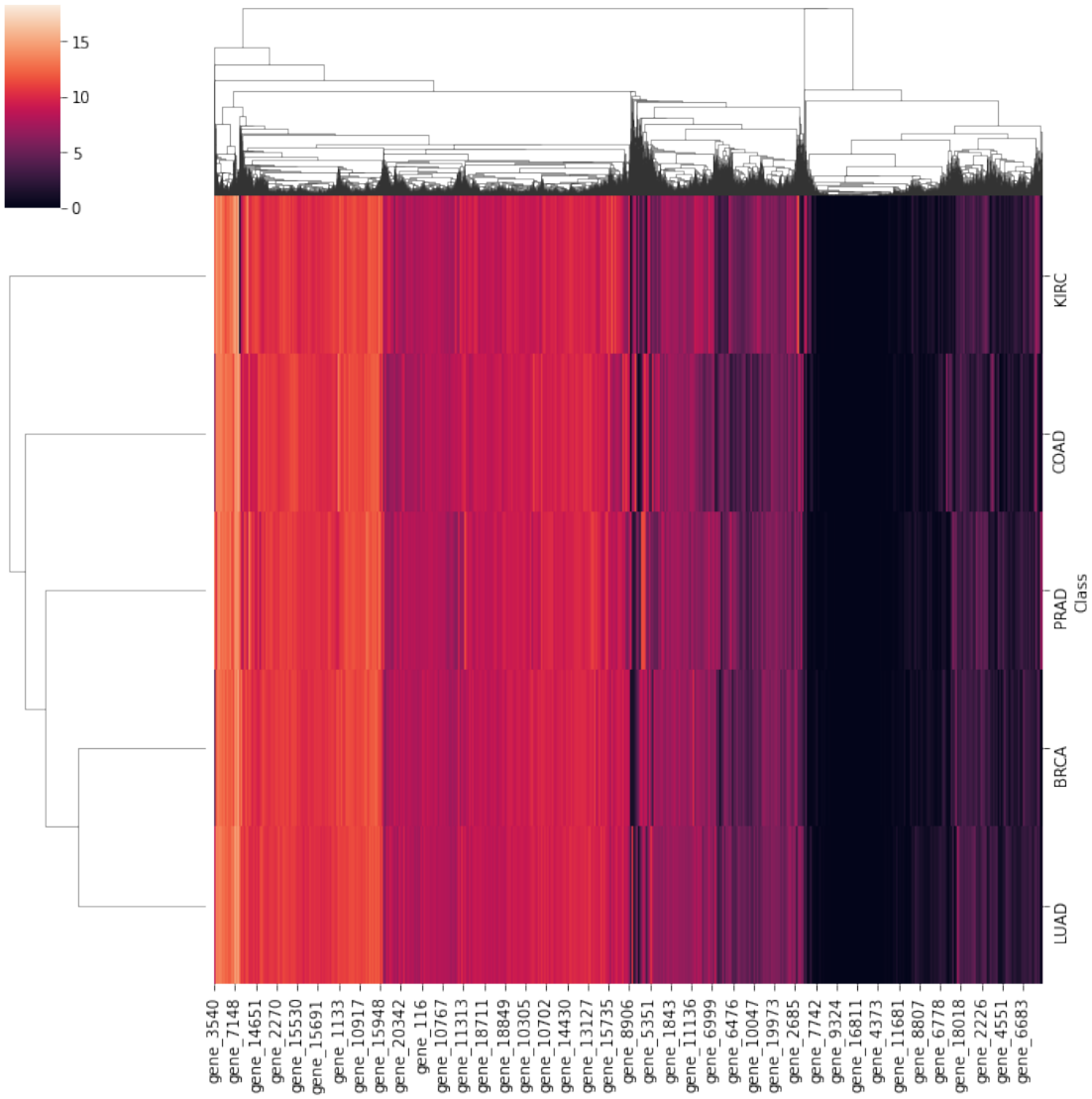
| Class | gene_10001 | gene_10002 | gene_10003 | gene_10004 | … | gene_9990 | \ |
|---|---|---|---|---|---|---|---|
| BRCA | 7.339461 | 7.900497 | 7.489146 | 7.508378 | … | 1.969278 | |
| COAD | 5.526673 | 7.487396 | 3.783493 | 6.959238 | … | 2.216178 | |
| KIRC | 6.895752 | 7.686932 | 7.269611 | 7.636246 | … | 1.824964 | |
| LUAD | 7.281878 | 7.041924 | 6.145042 | 7.148682 | … | 2.609490 | |
| PRAD | 7.946141 | 8.529695 | 5.696368 | 7.396572 | … | 1.623491 | |

| Class | gene_9991 | gene_9992 | gene_9993 | gene_9994 | gene_9995 | gene_9996 | \ |
|---|---|---|---|---|---|---|---|
| BRCA | 5.142237 | 1.736160 | 2.312551 | 1.696127 | 2.493789 | 0.046527 | |
| COAD | 0.354828 | 1.833606 | 1.619692 | 3.839205 | 2.396207 | 0.090327 | |
| KIRC | 0.596508 | 2.393303 | 1.872888 | 1.289448 | 3.139623 | 0.130416 | |
| LUAD | 2.801700 | 2.738326 | 1.869805 | 2.217144 | 2.459608 | 0.042070 | |
| PRAD | 4.594215 | 1.684084 | 2.588050 | 1.703772 | 3.568490 | 0.572893 | |

| Class | gene_9997 | gene_9998 | gene_9999 |
|---|---|---|---|
| BRCA | 2.099709 | 0.151063 | 6.954733 |
| COAD | 2.298246 | 0.065007 | 6.618466 |
| KIRC | 2.387948 | 0.148641 | 6.429343 |
| LUAD | 2.281828 | 0.056608 | 6.721517 |
| PRAD | 3.621548 | 0.094953 | 7.104225 |

[5 rows x 20531 columns]

[11]:
```
sns.clustermap(heatmap_data)
plt.savefig('heatmap_with_Seaborn_clustermap_python.jpg',dpi=150,␣
↪figsize=(8,12))
```




### 0.0.5 3.Perform Null-hypothesis testing.

[12]:
```
cat_data = merged_data.drop(['Unnamed: 0'], axis=1)
```

[13]:
```
cat_data
```

[13]:

| | Class | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | gene_6 | \ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | PRAD | 0.0 | 2.017209 | 3.265527 | 5.478487 | 10.431999 | 0.0 | 7.175175 | |
| 1 | LUAD | 0.0 | 0.592732 | 1.588421 | 7.586157 | 9.623011 | 0.0 | 6.816049 | |
| 2 | PRAD | 0.0 | 3.511759 | 4.327199 | 6.881787 | 9.870730 | 0.0 | 6.972130 | |
| 3 | PRAD | 0.0 | 3.663618 | 4.507649 | 6.659068 | 10.196184 | 0.0 | 7.843375 | |
| 4 | BRCA | 0.0 | 2.655741 | 2.821547 | 6.539454 | 9.738265 | 0.0 | 6.566967 | |
| .. | … | … | … | … | … | … | … | … | |
| 796 | BRCA | 0.0 | 1.865642 | 2.718197 | 7.350099 | 10.006003 | 0.0 | 6.764792 | |
| 797 | LUAD | 0.0 | 3.942955 | 4.453807 | 6.346597 | 10.056868 | 0.0 | 7.320331 | |
| 798 | COAD | 0.0 | 3.249582 | 3.707492 | 8.185901 | 9.504082 | 0.0 | 7.536589 | |
| 799 | PRAD | 0.0 | 2.590339 | 2.787976 | 7.318624 | 9.987136 | 0.0 | 9.213464 | |
| 800 | PRAD | 0.0 | 2.325242 | 3.805932 | 6.530246 | 9.560367 | 0.0 | 7.957027 | |

| | gene_7 | gene_8 | … | gene_20521 | gene_20522 | gene_20523 | gene_20524 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.591871 | 0.0 | … | 4.926711 | 8.210257 | 9.723516 | 7.220030 | |
| 1 | 0.000000 | 0.0 | … | 4.593372 | 7.323865 | 9.740931 | 6.256586 | |
| 2 | 0.452595 | 0.0 | … | 5.125213 | 8.127123 | 10.908640 | 5.401607 | |
| 3 | 0.434882 | 0.0 | … | 6.076566 | 8.792959 | 10.141520 | 8.942805 | |
| 4 | 0.360982 | 0.0 | … | 5.996032 | 8.891425 | 10.373790 | 7.181162 | |
| .. | … | … | … | … | … | … | … | |
| 796 | 0.496922 | 0.0 | … | 6.088133 | 9.118313 | 10.004852 | 4.484415 | |
| 797 | 0.000000 | 0.0 | … | 6.371876 | 9.623335 | 9.823921 | 6.555327 | |
| 798 | 1.811101 | 0.0 | … | 5.719386 | 8.610704 | 10.485517 | 3.589763 | |
| 799 | 0.000000 | 0.0 | … | 5.785237 | 8.605387 | 11.004677 | 4.745888 | |
| 800 | 0.000000 | 0.0 | … | 6.403075 | 8.594354 | 10.243079 | 9.139459 | |

| | gene_20525 | gene_20526 | gene_20527 | gene_20528 | gene_20529 | gene_20530 |
|---|---|---|---|---|---|---|
| 0 | 9.119813 | 12.003135 | 9.650743 | 8.921326 | 5.286759 | 0.000000 |
| 1 | 8.381612 | 12.674552 | 10.517059 | 9.397854 | 2.094168 | 0.000000 |
| 2 | 9.911597 | 9.045255 | 9.788359 | 10.090470 | 1.683023 | 0.000000 |
| 3 | 9.601208 | 11.392682 | 9.694814 | 9.684365 | 3.292001 | 0.000000 |
| 4 | 9.846910 | 11.922439 | 9.217749 | 9.461191 | 5.110372 | 0.000000 |
| .. | … | … | … | … | … | … |
| 796 | 9.614701 | 12.031267 | 9.813063 | 10.092770 | 8.819269 | 0.000000 |
| 797 | 9.064002 | 11.633422 | 10.317266 | 8.745983 | 9.659081 | 0.000000 |
| 798 | 9.350636 | 12.180944 | 10.681194 | 9.466711 | 4.677458 | 0.586693 |
| 799 | 9.626383 | 11.198279 | 10.335513 | 10.400581 | 5.718751 | 0.000000 |
| 800 | 10.102934 | 11.641081 | 10.607358 | 9.844794 | 4.550716 | 0.000000 |

[801 rows x 20532 columns]

[14]:
```
df_f_test=merged_data
```

[15]:
```
def f_test(df_f_test,gene):
    df_anova = df_f_test[[gene,'Class']]
    grps = pd.unique(df_anova.Class.values)
    grps
```

```
    d_data = {grp:df_anova[gene][df_anova.Class == grp] for grp in grps}
    F, p = stats.f_oneway(d_data['LUAD'], d_data['PRAD'], d_data['BRCA'],␣
 ↪d_data['KIRC'], d_data['COAD'])
    print("p_values:-",p)
    if p<0.05:
        print("reject null hypothesis")
    else:
        print("accept null hypothesis")

    return
```

[16]:
```
f_test(df_f_test,"gene_0")
```

```
p_values:- 0.07505540778266195
accept null hypothesis
```

[17]:
```
f_test(df_f_test,"gene_65")
```

```
p_values:- 3.406081969192749e-27
reject null hypothesis
```

[18]:
```
f_test(df_f_test,"gene_224")
```

```
p_values:- 1.7404736672399166e-61
reject null hypothesis
```

[19]:
```
f_test(df_f_test,"gene_5")
```

```
p_values:- nan
accept null hypothesis
```

[20]:
```
df_cat_data = merged_data
df_cat_data['Class'] = df_cat_data['Class'].map({'PRAD': 1, 'LUAD': 2, 'BRCA':␣
 ↪3, 'KIRC': 4, 'COAD': 5})
df_cat_data = df_cat_data.drop(['Unnamed: 0'],axis=1)
```

[21]:
```
from scipy.stats import shapiro
stat, p = shapiro(df_cat_data)
print('stat=%.2f' %(stat))
print('p=%.20f' %( p))

if p > 0.05:
    print('Normal Distribution')
else:
    print('Not Normal')
```

```
stat=0.92
p=0.00000000000000000000
Not Normal
```

[22]:
```
#K2 normality test
from scipy.stats import normaltest
k2_test = df_cat_data['Class']

stat, p = normaltest(k2_test)

print('stat=%.2f' %(stat))
print('p=%.20f' %( p))
if p > 0.05:
    print('Normal Distribution')
else:
    print('Not Normal')
```

```
stat=48.54
p=0.00000000002883341715
Not Normal
```

### 0.0.6 Dimensionality Reduction

[23]:
```
# Define data
df_pca = merged_data.drop(['Unnamed: 0'], axis=1)
df_pca = df_pca.drop(['Class'], axis=1)
df_pca.head()
```

[23]:

| | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | gene_6 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 2.017209 | 3.265527 | 5.478487 | 10.431999 | 0.0 | 7.175175 | |
| 1 | 0.0 | 0.592732 | 1.588421 | 7.586157 | 9.623011 | 0.0 | 6.816049 | |
| 2 | 0.0 | 3.511759 | 4.327199 | 6.881787 | 9.870730 | 0.0 | 6.972130 | |
| 3 | 0.0 | 3.663618 | 4.507649 | 6.659068 | 10.196184 | 0.0 | 7.843375 | |
| 4 | 0.0 | 2.655741 | 2.821547 | 6.539454 | 9.738265 | 0.0 | 6.566967 | |

| | gene_7 | gene_8 | gene_9 | … | gene_20521 | gene_20522 | gene_20523 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.591871 | 0.0 | 0.0 | … | 4.926711 | 8.210257 | 9.723516 | |
| 1 | 0.000000 | 0.0 | 0.0 | … | 4.593372 | 7.323865 | 9.740931 | |
| 2 | 0.452595 | 0.0 | 0.0 | … | 5.125213 | 8.127123 | 10.908640 | |
| 3 | 0.434882 | 0.0 | 0.0 | … | 6.076566 | 8.792959 | 10.141520 | |
| 4 | 0.360982 | 0.0 | 0.0 | … | 5.996032 | 8.891425 | 10.373790 | |

| | gene_20524 | gene_20525 | gene_20526 | gene_20527 | gene_20528 | gene_20529 | \ |
|---|---|---|---|---|---|---|---|
| 0 | 7.220030 | 9.119813 | 12.003135 | 9.650743 | 8.921326 | 5.286759 | |
| 1 | 6.256586 | 8.381612 | 12.674552 | 10.517059 | 9.397854 | 2.094168 | |
| 2 | 5.401607 | 9.911597 | 9.045255 | 9.788359 | 10.090470 | 1.683023 | |

```
3    8.942805    9.601208   11.392682    9.694814    9.684365    3.292001
4    7.181162    9.846910   11.922439    9.217749    9.461191    5.110372

   gene_20530
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0

[5 rows x 20531 columns]
```

[24]:
```
df_pca.values.shape
```

[24]:
```
(801, 20531)
```

[25]:
```
x_pca = df_pca.values
```

### 0.0.7 Scaling the data using standard scaler method

[26]:
```
scaler = StandardScaler()
X_Scaled = scaler.fit_transform(x_pca)
X_Scaled
```

[26]:
```
array([[-0.19479935, -0.82802988,  0.15980044, …, -1.18793812,
        -0.11648251, -0.26190144],
       [-0.19479935, -2.01501735, -1.415042  , …, -0.34227662,
        -1.65688871, -0.26190144],
       [-0.19479935,  0.41734754,  1.15673547, …,  0.88686027,
        -1.85526414, -0.26190144],
       …,
       [-0.19479935,  0.19888076,  0.57481583, …, -0.22008186,
        -0.41046699,  1.3485582 ],
       [-0.19479935, -0.35045311, -0.28863152, …,  1.43719268,
         0.09195083, -0.26190144],
       [-0.19479935, -0.57135218,  0.66725377, …,  0.45087581,
        -0.47161901, -0.26190144]])
```

### 0.0.8 Perform PCA with n_components=2

[27]:
```
#define the n_components as 2
pca_with_2=PCA(n_components=2)
```

[28]:
```
#Perform fit transform on the scaled data
X_pca_with_2 = pca_with_2.fit_transform(X_Scaled)
X_pca_with_2.shape
```

[28]:
```
(801, 2)
```

[29]:
```
X_pca_with_2
```

[29]:
```
array([[-57.44698689,  95.41098072],
       [-16.91943009,   0.73247023],
       [-70.34521806, -19.30332628],
       …,
       [ -4.13308983,  15.69001452],
       [-30.81475747,  33.52642254],
       [-22.34455665,   4.05235625]])
```

[30]:
```
# Put the data back on the 2 columns defined
df_pca = pd.DataFrame(X_pca_with_2)
df_pca.columns = ['pca1','pca2']

# Add the convereted categorical data for
df_pca['cancer_type']=df_cat_data['Class']
df_pca
```

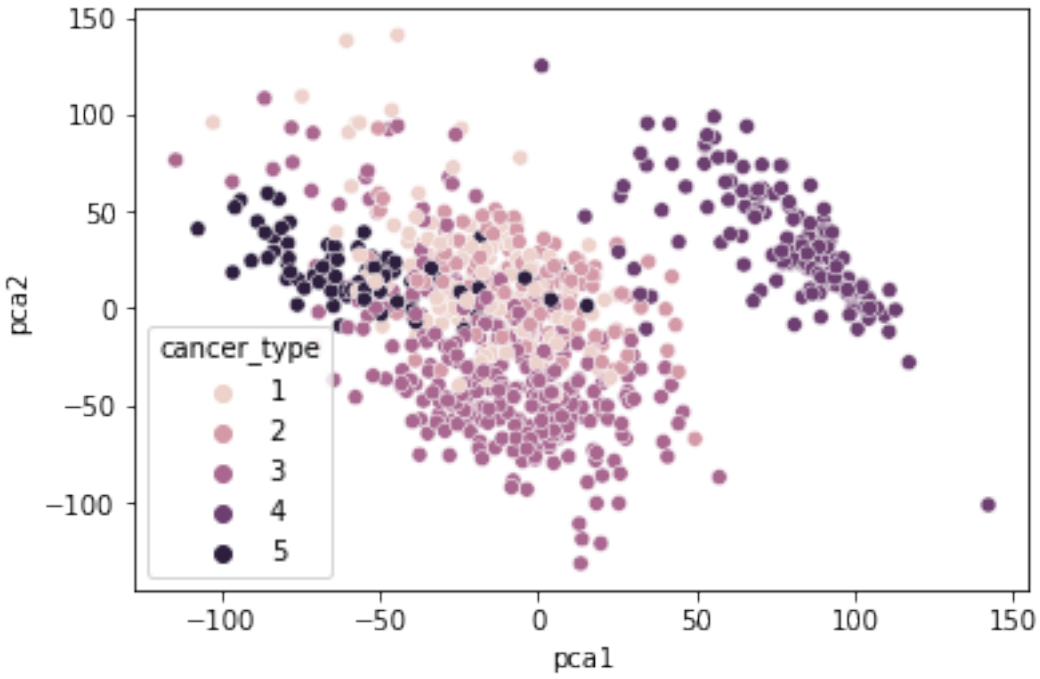
[30]:

| | pca1 | pca2 | cancer_type |
|---|---|---|---|
| 0 | -57.446987 | 95.410981 | 1 |
| 1 | -16.919430 | 0.732470 | 2 |
| 2 | -70.345218 | -19.303326 | 1 |
| 3 | -49.161591 | -9.227586 | 1 |
| 4 | -18.132534 | -51.327797 | 3 |
| .. | … | … | … |
| 796 | -12.417385 | -42.321573 | 3 |
| 797 | -29.415555 | 28.526282 | 2 |
| 798 | -4.133090 | 15.690015 | 5 |
| 799 | -30.814757 | 33.526423 | 1 |
| 800 | -22.344557 | 4.052356 | 1 |

[801 rows x 3 columns]

[31]:
```
# Present the data on the 5 clusters using seaborn maps
sns.scatterplot(x='pca1',y='pca2', hue = 'cancer_type',data=df_pca)
```

[31]:
```
<AxesSubplot:xlabel='pca1', ylabel='pca2'>
```




### 0.0.9 PCA with n_components=.995

[32]:
```
pca_with_995=PCA(.995)
X_pca_with_995 = pca_with_995.fit_transform(x_pca)
X_pca_with_995.shape
X_pca_with_995
```

[32]:
```
array([[-6.27554152e+01, -9.40719735e+01,  8.95198311e+01, …,
         3.09258084e+00,  7.13597730e-01, -8.21221710e-02],
       [-2.43289636e+00,  9.05858418e+01, -1.06730787e+00, …,
         1.39674724e-02, -3.95175744e-01, -9.49947250e-01],
       [-7.12668528e+01, -8.06460774e+00,  6.61124549e+01, …,
         1.28898532e-01, -2.64530262e-01,  3.84594189e-01],
       …,
       [ 1.04862615e+01,  2.15705946e+01,  4.13458784e+01, …,
        -6.47882986e-01, -2.07256774e-01,  1.38942922e-01],
       [-5.50636049e+01, -9.23947780e+01,  8.00500394e+01, …,
         1.74673062e+00,  2.02232239e+00, -1.92708948e+00],
       [-4.91030338e+01, -5.09976391e+01,  4.05037544e+01, …,
         1.80367340e+00,  2.22994027e+00, -8.07255452e-01]])
```

[33]:
```
df_pca_995 = pd.DataFrame(X_pca_with_995)
df_pca_995['cancer_type']=df_cat_data['Class']
```

```
df_pca_995
```

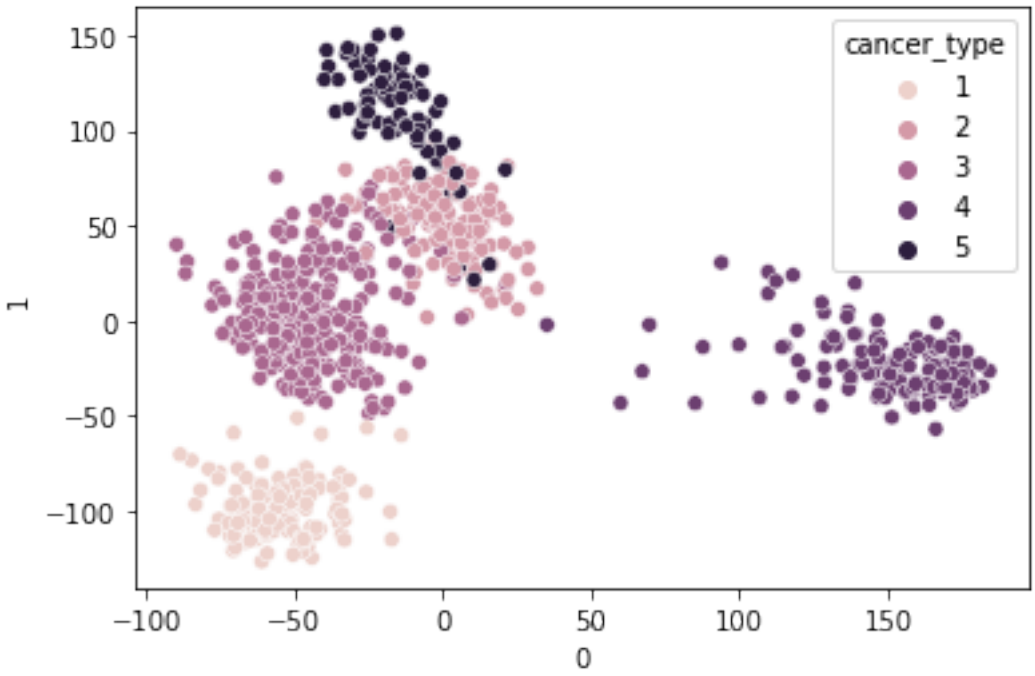
[33]:

```
             0          1          2          3          4           5  \
0   -62.755415 -94.071973  89.519831 -15.942567  81.423539  -13.998292
1    -2.432896  90.585842  -1.067308 -53.083120 -15.676684   60.842472
2   -71.266853  -8.064608  66.112455  81.381475  -7.525685  109.824273
3   -84.770785 -73.244566  74.181000  27.022697 -18.044895   50.116433
4   -69.560171  -9.612940 -67.497549  34.868543  -1.795849   -6.676780
..         …          …          …          …          …           …
796 -60.861882 -22.278633 -80.927167  42.670292   7.843763   -4.545218
797 -14.465433  53.392194  38.153904 -63.217345  22.799082   39.543441
798  10.486261  21.570595  41.345878 -59.639929  -2.163066  -96.453878
799 -55.063605 -92.394778  80.050039  -7.782015  15.180574    2.563620
800 -49.103034 -50.997639  40.503754 -31.495505 -10.361908   -1.272555

             6          7          8          9  …       738       739  \
0     7.716073 -22.936551 -32.837892  -2.202680  … -4.081064 -0.626193
1    10.257369 -48.822959  14.257400 -12.214352  …  0.215619 -0.593678
2     5.519407 -13.364480  38.415728  -5.124731  …  0.263786  0.328453
3    -3.495197 -11.318520   8.319656  -3.149509  …  0.381578  0.652455
4    -2.840781  16.780157 -49.319753  10.508631  …  1.488047  2.767486
..         …          …          …          …  …         …         …
796 -27.602910  -8.840676 -31.531870   6.380236  … -0.780676  0.105227
797 -47.899401  39.925172 -12.413483  43.364820  … -0.712822  0.624739
798  38.375897  46.997294  60.604643  59.967025  …  0.269628 -0.348648
799   8.487660  10.571657  11.710577   1.304005  …  0.045885 -2.222754
800   9.185948 -31.629661  40.799717  -5.265109  … -1.429271 -1.286569

          740       741       742       743       744       745       746  \
0   -1.265756 -0.017984 -2.740860  0.944037  3.092581  0.713598 -0.082122
1   -0.403462  1.181537  0.490910  0.197768  0.013967 -0.395176 -0.949947
2    0.004078  0.363928 -1.109210  0.331488  0.128899 -0.264530  0.384594
3   -3.624900 -1.203028 -2.347912  1.577992 -0.781748  0.120442 -0.057973
4   -0.631562 -0.794275 -0.514008 -1.875969 -2.526109 -1.073803 -1.161728
..        …         …         …         …         …         …         …
796 -2.001001  1.579115  0.955344  0.085881  2.667448  0.632850  0.023523
797 -0.162403 -0.238540  0.584705  1.404867  0.564251 -0.054682 -0.905574
798 -0.531710  0.055553  0.220559  0.331122 -0.647883 -0.207257  0.138943
799 -4.115667 -0.064646 -0.447662 -0.243658  1.746731  2.022322 -1.927089
800 -0.166544  3.095998  0.935408  2.854994  1.803673  2.229940 -0.807255

     cancer_type
0              1
1              2
2              1
3              1
4              3
```

```
..          …
796           3
797           2
798           5
799           1
800           1

[801 rows x 748 columns]
```

[34]:
```
sns.scatterplot(x=0,y=1,hue = 'cancer_type', data=df_pca_995)
```

[34]:
```
<AxesSubplot:xlabel='0', ylabel='1'>
```




### 0.0.10 Dimensionality reduction using TSNE

[35]:
```
df_tsne_data = merged_data
non_num = ['Unnamed: 0','Class']
df_tsne_data = df_tsne_data.drop(non_num, axis=1)
df_tsne_data
```

[35]:
```
     gene_0    gene_1    gene_2    gene_3     gene_4  gene_5    gene_6  \
0       0.0  2.017209  3.265527  5.478487  10.431999     0.0  7.175175
1       0.0  0.592732  1.588421  7.586157   9.623011     0.0  6.816049
```

```
2       0.0  3.511759  4.327199  6.881787   9.870730     0.0  6.972130
3       0.0  3.663618  4.507649  6.659068  10.196184     0.0  7.843375
4       0.0  2.655741  2.821547  6.539454   9.738265     0.0  6.566967
..      …       …        …        …         …      …       …
796     0.0  1.865642  2.718197  7.350099  10.006003     0.0  6.764792
797     0.0  3.942955  4.453807  6.346597  10.056868     0.0  7.320331
798     0.0  3.249582  3.707492  8.185901   9.504082     0.0  7.536589
799     0.0  2.590339  2.787976  7.318624   9.987136     0.0  9.213464
800     0.0  2.325242  3.805932  6.530246   9.560367     0.0  7.957027

       gene_7  gene_8  gene_9  …  gene_20521  gene_20522  gene_20523  \
0    0.591871     0.0     0.0  …    4.926711    8.210257    9.723516
1    0.000000     0.0     0.0  …    4.593372    7.323865    9.740931
2    0.452595     0.0     0.0  …    5.125213    8.127123   10.908640
3    0.434882     0.0     0.0  …    6.076566    8.792959   10.141520
4    0.360982     0.0     0.0  …    5.996032    8.891425   10.373790
..        …      …       …   …  …         …           …          …
796  0.496922     0.0     0.0  …    6.088133    9.118313   10.004852
797  0.000000     0.0     0.0  …    6.371876    9.623335    9.823921
798  1.811101     0.0     0.0  …    5.719386    8.610704   10.485517
799  0.000000     0.0     0.0  …    5.785237    8.605387   11.004677
800  0.000000     0.0     0.0  …    6.403075    8.594354   10.243079

     gene_20524  gene_20525  gene_20526  gene_20527  gene_20528  gene_20529  \
0      7.220030    9.119813   12.003135    9.650743    8.921326    5.286759
1      6.256586    8.381612   12.674552   10.517059    9.397854    2.094168
2      5.401607    9.911597    9.045255    9.788359   10.090470    1.683023
3      8.942805    9.601208   11.392682    9.694814    9.684365    3.292001
4      7.181162    9.846910   11.922439    9.217749    9.461191    5.110372
..          …           …          …           …           …           …
796    4.484415    9.614701   12.031267    9.813063   10.092770    8.819269
797    6.555327    9.064002   11.633422   10.317266    8.745983    9.659081
798    3.589763    9.350636   12.180944   10.681194    9.466711    4.677458
799    4.745888    9.626383   11.198279   10.335513   10.400581    5.718751
800    9.139459   10.102934   11.641081   10.607358    9.844794    4.550716

     gene_20530
0      0.000000
1      0.000000
2      0.000000
3      0.000000
4      0.000000
..          …
796    0.000000
797    0.000000
798    0.586693
799    0.000000
```

```
800    0.000000

[801 rows x 20531 columns]
```

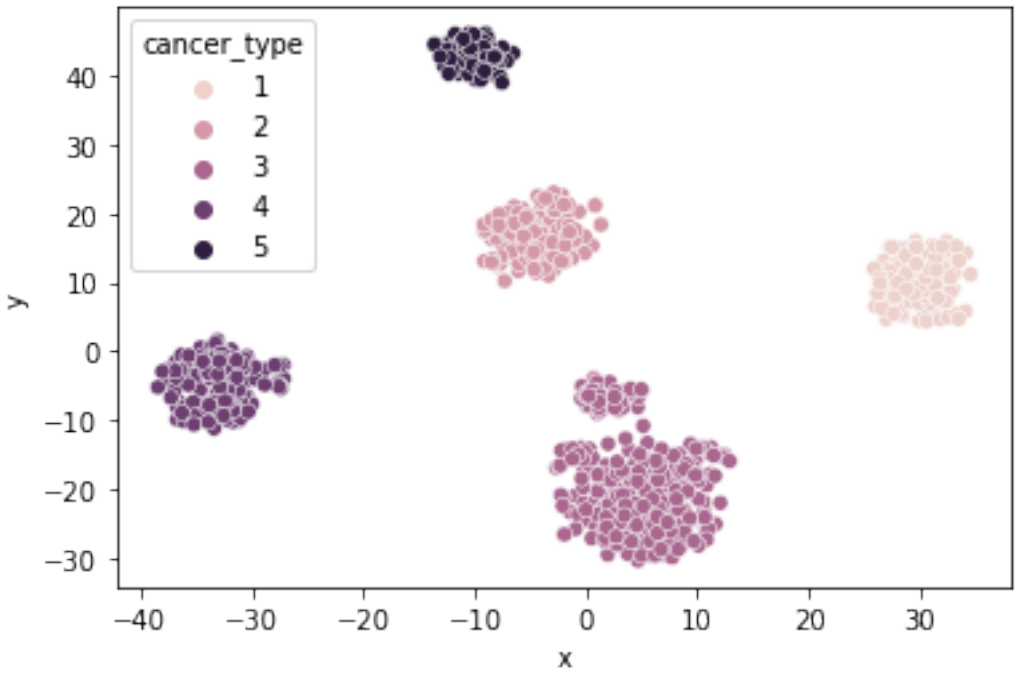
[36]:
```
#import T-SNE from sklearn
m = TSNE(learning_rate=50)
```

[37]:
```
tnse_features = m.fit_transform(df_tsne_data)
tnse_features[1:4,:]
```

[37]:
```
array([[-6.3689394, 13.379185 ],
       [29.59662  , 16.001747 ],
       [29.51575  , 15.109866 ]], dtype=float32)
```

[38]:
```
df_tsne_data['x'] = tnse_features[:,0]
df_tsne_data['y'] = tnse_features[:,1]
```

[39]:
```
df_tsne_data['cancer_type']=df_cat_data['Class']
sns.scatterplot(x='x',y='y',hue = 'cancer_type', data=df_tsne_data)
plt.show()
```



### 0.0.11 Dimensionality reduction using LDA

[40]:
```
df_lda = merged_data.drop(['Unnamed: 0'], axis=1)
df_lda = df_lda.drop(['Class'], axis=1)
x_lda = df_lda
x_lda
```

[40]:

| | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | gene_6 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 2.017209 | 3.265527 | 5.478487 | 10.431999 | 0.0 | 7.175175 | |
| 1 | 0.0 | 0.592732 | 1.588421 | 7.586157 | 9.623011 | 0.0 | 6.816049 | |
| 2 | 0.0 | 3.511759 | 4.327199 | 6.881787 | 9.870730 | 0.0 | 6.972130 | |
| 3 | 0.0 | 3.663618 | 4.507649 | 6.659068 | 10.196184 | 0.0 | 7.843375 | |
| 4 | 0.0 | 2.655741 | 2.821547 | 6.539454 | 9.738265 | 0.0 | 6.566967 | |
| .. | … | … | … | … | … | … | … | |
| 796 | 0.0 | 1.865642 | 2.718197 | 7.350099 | 10.006003 | 0.0 | 6.764792 | |
| 797 | 0.0 | 3.942955 | 4.453807 | 6.346597 | 10.056868 | 0.0 | 7.320331 | |
| 798 | 0.0 | 3.249582 | 3.707492 | 8.185901 | 9.504082 | 0.0 | 7.536589 | |
| 799 | 0.0 | 2.590339 | 2.787976 | 7.318624 | 9.987136 | 0.0 | 9.213464 | |
| 800 | 0.0 | 2.325242 | 3.805932 | 6.530246 | 9.560367 | 0.0 | 7.957027 | |

| | gene_7 | gene_8 | gene_9 | … | gene_20521 | gene_20522 | gene_20523 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.591871 | 0.0 | 0.0 | … | 4.926711 | 8.210257 | 9.723516 | |
| 1 | 0.000000 | 0.0 | 0.0 | … | 4.593372 | 7.323865 | 9.740931 | |
| 2 | 0.452595 | 0.0 | 0.0 | … | 5.125213 | 8.127123 | 10.908640 | |
| 3 | 0.434882 | 0.0 | 0.0 | … | 6.076566 | 8.792959 | 10.141520 | |
| 4 | 0.360982 | 0.0 | 0.0 | … | 5.996032 | 8.891425 | 10.373790 | |
| .. | … | … | … | … | … | … | … | |
| 796 | 0.496922 | 0.0 | 0.0 | … | 6.088133 | 9.118313 | 10.004852 | |
| 797 | 0.000000 | 0.0 | 0.0 | … | 6.371876 | 9.623335 | 9.823921 | |
| 798 | 1.811101 | 0.0 | 0.0 | … | 5.719386 | 8.610704 | 10.485517 | |
| 799 | 0.000000 | 0.0 | 0.0 | … | 5.785237 | 8.605387 | 11.004677 | |
| 800 | 0.000000 | 0.0 | 0.0 | … | 6.403075 | 8.594354 | 10.243079 | |

| | gene_20524 | gene_20525 | gene_20526 | gene_20527 | gene_20528 | gene_20529 | \ |
|---|---|---|---|---|---|---|---|
| 0 | 7.220030 | 9.119813 | 12.003135 | 9.650743 | 8.921326 | 5.286759 | |
| 1 | 6.256586 | 8.381612 | 12.674552 | 10.517059 | 9.397854 | 2.094168 | |
| 2 | 5.401607 | 9.911597 | 9.045255 | 9.788359 | 10.090470 | 1.683023 | |
| 3 | 8.942805 | 9.601208 | 11.392682 | 9.694814 | 9.684365 | 3.292001 | |
| 4 | 7.181162 | 9.846910 | 11.922439 | 9.217749 | 9.461191 | 5.110372 | |
| .. | … | … | … | … | … | … | |
| 796 | 4.484415 | 9.614701 | 12.031267 | 9.813063 | 10.092770 | 8.819269 | |
| 797 | 6.555327 | 9.064002 | 11.633422 | 10.317266 | 8.745983 | 9.659081 | |
| 798 | 3.589763 | 9.350636 | 12.180944 | 10.681194 | 9.466711 | 4.677458 | |
| 799 | 4.745888 | 9.626383 | 11.198279 | 10.335513 | 10.400581 | 5.718751 | |
| 800 | 9.139459 | 10.102934 | 11.641081 | 10.607358 | 9.844794 | 4.550716 | |

| | gene_20530 |
|---|---|

```
0      0.000000
1      0.000000
2      0.000000
3      0.000000
4      0.000000
..          …
796    0.000000
797    0.000000
798    0.586693
799    0.000000
800    0.000000

[801 rows x 20531 columns]
```

[41]:
```
x_lda.shape
```

[41]:
```
(801, 20531)
```

[42]:
```
y_lda = merged_data['Class']
y_lda.values
```

[42]:
```
array([1, 2, 1, 1, 3, 1, 4, 1, 3, 1, 3, 4, 1, 3, 3, 3, 2, 4, 4, 1, 3, 4,
       2, 3, 4, 2, 5, 3, 3, 3, 3, 3, 4, 3, 1, 3, 4, 2, 3, 3, 4, 1, 1, 4,
       4, 3, 1, 5, 3, 2, 3, 2, 3, 1, 5, 3, 3, 5, 4, 3, 2, 4, 3, 2, 1, 5,
       3, 1, 4, 3, 4, 3, 3, 2, 3, 2, 3, 4, 1, 5, 3, 1, 3, 3, 1, 1, 3, 3,
       4, 3, 1, 1, 3, 3, 3, 1, 5, 3, 1, 3, 3, 4, 3, 4, 2, 4, 2, 5, 2, 2,
       1, 3, 2, 1, 3, 4, 4, 4, 3, 3, 2, 4, 2, 3, 1, 1, 1, 3, 4, 2, 5, 3,
       5, 3, 3, 4, 2, 3, 4, 5, 3, 1, 3, 4, 2, 5, 1, 3, 2, 2, 2, 2, 3, 3,
       2, 3, 3, 1, 1, 2, 1, 2, 4, 3, 1, 2, 5, 4, 2, 3, 4, 2, 3, 2, 3, 3,
       3, 1, 3, 4, 5, 4, 3, 1, 1, 1, 2, 2, 3, 2, 2, 4, 2, 1, 2, 3, 3, 3,
       2, 2, 3, 4, 4, 4, 4, 1, 3, 1, 3, 2, 2, 3, 1, 3, 1, 3, 3, 3, 2, 3,
       4, 2, 4, 4, 2, 3, 4, 1, 3, 2, 2, 1, 5, 3, 4, 1, 4, 5, 3, 4, 4, 2,
       1, 1, 2, 2, 4, 3, 3, 5, 3, 1, 5, 3, 1, 4, 1, 1, 1, 3, 5, 5, 2, 5,
       5, 1, 2, 3, 3, 4, 4, 3, 5, 1, 4, 1, 3, 3, 4, 3, 3, 3, 3, 2, 2, 3,
       3, 3, 4, 4, 4, 4, 3, 3, 3, 4, 3, 3, 2, 1, 3, 3, 5, 2, 1, 3, 3, 3,
       5, 3, 1, 3, 5, 2, 2, 1, 4, 3, 4, 4, 2, 5, 4, 3, 3, 3, 3, 4, 3, 3,
       1, 3, 4, 3, 2, 1, 4, 3, 1, 5, 3, 3, 3, 2, 2, 2, 3, 3, 1, 2, 3, 4,
       3, 5, 5, 2, 3, 4, 3, 3, 3, 5, 2, 5, 4, 1, 4, 3, 3, 4, 3, 5, 1, 2,
       1, 3, 4, 1, 3, 5, 4, 4, 5, 5, 1, 3, 3, 5, 4, 3, 1, 3, 3, 3, 2, 2,
       4, 2, 3, 4, 5, 1, 3, 2, 1, 3, 3, 3, 2, 3, 3, 1, 3, 1, 5, 3, 2, 3,
       3, 2, 3, 3, 3, 4, 2, 1, 3, 1, 4, 3, 4, 5, 3, 1, 2, 4, 3, 3, 4, 3,
       2, 3, 3, 1, 5, 3, 4, 2, 1, 3, 1, 3, 3, 3, 3, 4, 2, 3, 4, 3, 3, 2,
       2, 4, 5, 1, 5, 3, 4, 4, 3, 1, 4, 5, 2, 2, 3, 1, 1, 3, 1, 2, 4, 1,
       3, 2, 1, 2, 3, 3, 5, 2, 4, 5, 2, 3, 1, 3, 3, 1, 3, 5, 3, 5, 4, 3,
       3, 2, 2, 2, 5, 4, 2, 2, 3, 3, 4, 1, 2, 1, 3, 4, 3, 4, 4, 1, 1, 2,
       3, 4, 5, 5, 3, 4, 4, 3, 3, 1, 4, 5, 3, 3, 5, 2, 3, 3, 3, 4, 1, 2,
       2, 3, 4, 5, 4, 4, 3, 1, 2, 4, 3, 5, 2, 2, 2, 1, 2, 4, 3, 3, 5, 1,
```

```
3, 3, 3, 4, 2, 2, 3, 1, 2, 2, 3, 4, 1, 5, 2, 1, 5, 2, 5, 4, 4, 3,
3, 4, 4, 5, 3, 2, 1, 1, 4, 3, 2, 3, 3, 5, 3, 1, 1, 3, 3, 5, 3, 4,
3, 3, 5, 3, 1, 3, 3, 4, 1, 2, 3, 3, 4, 3, 3, 3, 3, 3, 5, 2, 2, 3,
3, 3, 1, 3, 3, 4, 2, 2, 4, 4, 2, 4, 5, 3, 5, 4, 3, 3, 1, 1, 1, 2,
1, 1, 5, 3, 3, 5, 2, 4, 3, 4, 5, 3, 3, 3, 1, 2, 4, 3, 1, 4, 1, 3,
2, 4, 1, 2, 1, 1, 3, 4, 1, 2, 5, 5, 3, 3, 3, 2, 4, 4, 4, 3, 2, 4,
1, 3, 2, 1, 3, 1, 3, 4, 3, 5, 1, 1, 4, 1, 4, 3, 2, 2, 3, 3, 4, 3,
3, 3, 3, 4, 4, 1, 5, 4, 3, 3, 4, 3, 2, 3, 3, 1, 3, 1, 3, 5, 3, 3,
4, 2, 3, 3, 1, 3, 1, 5, 3, 3, 3, 2, 3, 2, 3, 2, 4, 4, 5, 5, 3, 3,
3, 2, 3, 2, 4, 3, 2, 4, 2, 2, 2, 4, 3, 1, 3, 3, 4, 3, 1, 4, 3, 3,
3, 3, 1, 2, 3, 2, 5, 1, 1])
```

[43]:
```
lda = LDA(n_components=2)
x_r2 = lda.fit(x_lda,y_lda).transform(x_lda)
```

[44]:
```
lda.explained_variance_ratio_
```

[44]:
```
array([0.36219022, 0.30156109])
```

[45]:
```
x_r3 = pd.DataFrame(data=x_r2)
x_r3['y']=y_lda
x_r3
```

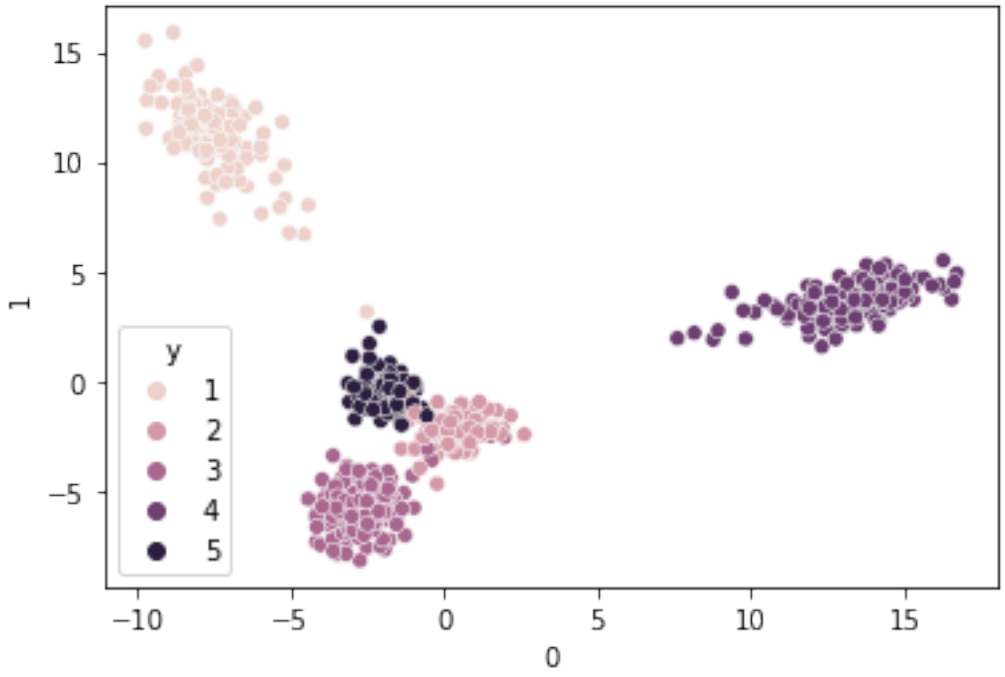
[45]:

| | 0 | 1 | y |
|---|---|---|---|
| 0 | -7.958125 | 10.922818 | 1 |
| 1 | -0.301563 | -2.780898 | 2 |
| 2 | -6.424952 | 8.870978 | 1 |
| 3 | -6.934259 | 10.417199 | 1 |
| 4 | -2.872004 | -4.912284 | 3 |
| .. | … | … | .. |
| 796 | -2.491183 | -6.516482 | 3 |
| 797 | 0.217789 | -1.859410 | 2 |
| 798 | -1.426674 | -0.474514 | 5 |
| 799 | -7.800641 | 12.104337 | 1 |
| 800 | -7.306312 | 7.388476 | 1 |

[801 rows x 3 columns]

[46]:
```
sns.scatterplot(x=0,y=1,hue = 'y', data=x_r3)
```

[46]:
```
<AxesSubplot:xlabel='0', ylabel='1'>
```




### 0.0.12 Clustering Genes and Samples:

**KMEANS Clustering with PCA = 2**

[47]:
```
clusters = KMeans(5, n_init = 5)
clusters.fit(X_pca_with_2)

clusters.labels_
```

[47]:
```
array([4, 0, 1, 1, 2, 1, 3, 0, 2, 1, 0, 3, 0, 4, 2, 2, 0, 3, 3, 1, 2, 0,
       0, 1, 3, 0, 1, 2, 1, 2, 2, 2, 3, 4, 1, 2, 3, 0, 1, 4, 3, 1, 0, 3,
       3, 2, 4, 1, 2, 0, 2, 1, 2, 1, 1, 1, 2, 1, 3, 0, 0, 0, 2, 0, 1, 1,
       0, 1, 3, 1, 3, 0, 1, 0, 2, 1, 2, 4, 0, 1, 2, 1, 2, 2, 4, 0, 2, 1,
       3, 1, 1, 1, 2, 4, 2, 1, 1, 2, 2, 0, 2, 3, 2, 3, 0, 3, 1, 1, 1, 2,
       4, 2, 0, 1, 2, 3, 3, 3, 2, 2, 0, 3, 1, 1, 0, 0, 1, 2, 3, 1, 1, 1,
       4, 1, 2, 3, 3, 2, 3, 1, 2, 1, 2, 3, 0, 4, 4, 2, 0, 1, 0, 0, 2, 2,
       0, 0, 2, 1, 1, 0, 4, 0, 3, 0, 0, 0, 1, 3, 0, 2, 3, 0, 2, 1, 2, 1,
       0, 0, 2, 3, 1, 3, 2, 0, 1, 0, 0, 0, 2, 0, 0, 3, 4, 2, 0, 2, 4, 2,
       1, 0, 2, 3, 3, 3, 3, 2, 4, 1, 0, 0, 0, 2, 0, 1, 0, 2, 2, 2, 0, 0,
       3, 0, 3, 3, 0, 2, 3, 1, 2, 0, 4, 2, 1, 4, 3, 0, 3, 1, 0, 3, 3, 2,
       1, 0, 0, 1, 3, 0, 2, 1, 2, 1, 0, 2, 0, 3, 4, 0, 0, 2, 0, 1, 2, 1,
       1, 1, 2, 1, 2, 3, 3, 2, 1, 4, 3, 1, 2, 2, 3, 2, 2, 2, 4, 1, 0, 2,
       2, 2, 3, 3, 3, 3, 4, 2, 2, 3, 2, 2, 0, 0, 0, 2, 4, 1, 1, 2, 2, 2,
       4, 0, 2, 2, 1, 1, 0, 1, 3, 1, 3, 3, 2, 1, 0, 2, 2, 2, 2, 3, 2, 2,
```

```
       1, 1, 3, 2, 0, 0, 3, 2, 1, 1, 0, 2, 2, 0, 4, 0, 2, 2, 0, 0, 2, 3,
       2, 1, 0, 0, 2, 3, 2, 2, 1, 1, 0, 1, 0, 1, 3, 0, 2, 3, 2, 1, 1, 0,
       1, 4, 3, 1, 2, 1, 3, 3, 1, 1, 4, 2, 4, 1, 3, 4, 0, 1, 2, 2, 0, 0,
       3, 0, 2, 3, 1, 2, 2, 1, 0, 1, 4, 2, 1, 2, 1, 0, 1, 1, 1, 2, 2, 2,
       2, 0, 2, 0, 2, 3, 0, 2, 2, 0, 3, 1, 3, 4, 2, 1, 0, 3, 2, 2, 3, 2,
       0, 1, 2, 1, 4, 1, 3, 0, 4, 1, 1, 2, 2, 2, 2, 3, 2, 0, 3, 2, 2, 1,
       0, 3, 1, 0, 1, 0, 3, 3, 1, 0, 3, 1, 0, 1, 2, 0, 0, 0, 2, 2, 3, 0,
       0, 4, 2, 4, 2, 2, 1, 1, 3, 1, 1, 1, 1, 4, 0, 0, 2, 1, 2, 4, 3, 2,
       2, 1, 2, 2, 1, 3, 0, 1, 2, 2, 0, 0, 1, 0, 2, 3, 1, 3, 3, 4, 0, 0,
       0, 3, 1, 1, 2, 3, 3, 2, 0, 0, 3, 1, 0, 2, 1, 0, 2, 0, 2, 3, 2, 1,
       0, 1, 3, 1, 3, 3, 1, 0, 1, 3, 2, 1, 4, 0, 0, 0, 0, 3, 2, 2, 1, 1,
       2, 0, 2, 3, 0, 1, 1, 1, 1, 2, 2, 3, 1, 4, 1, 0, 1, 0, 4, 3, 3, 2,
       2, 3, 3, 1, 2, 0, 1, 4, 3, 2, 0, 2, 0, 1, 0, 4, 0, 0, 2, 4, 2, 3,
       2, 0, 0, 2, 0, 1, 2, 3, 1, 0, 2, 2, 3, 0, 2, 2, 2, 4, 1, 0, 0, 1,
       2, 2, 0, 2, 2, 3, 1, 2, 3, 3, 0, 3, 1, 0, 4, 3, 2, 2, 0, 0, 1, 0,
       1, 2, 1, 2, 2, 1, 1, 3, 2, 3, 1, 2, 2, 0, 2, 0, 3, 2, 2, 3, 1, 2,
       0, 3, 1, 0, 0, 0, 4, 3, 1, 0, 1, 1, 2, 0, 2, 0, 3, 3, 3, 1, 0, 3,
       0, 2, 0, 4, 2, 1, 2, 3, 1, 0, 0, 0, 3, 0, 3, 2, 0, 0, 2, 4, 3, 2,
       1, 2, 1, 3, 3, 0, 1, 3, 1, 4, 3, 2, 0, 2, 1, 2, 4, 1, 1, 1, 0, 2,
       3, 0, 2, 2, 0, 2, 1, 0, 2, 0, 2, 0, 2, 1, 0, 0, 3, 3, 1, 1, 2, 2,
       2, 1, 2, 0, 0, 0, 0, 3, 1, 0, 1, 3, 2, 4, 1, 1, 3, 2, 2, 0, 2, 2,
       2, 2, 2, 0, 2, 1, 0, 1, 1], dtype=int32)
```

[48]:
```
pca_2_data = pd.DataFrame(data=X_pca_with_2,columns=['pca1','pca2'])
pca_2_data.head()
```

[48]:

| | pca1 | pca2 |
|---|---|---|
| 0 | -57.446987 | 95.410981 |
| 1 | -16.919430 | 0.732470 |
| 2 | -70.345218 | -19.303326 |
| 3 | -49.161591 | -9.227586 |
| 4 | -18.132534 | -51.327797 |

[49]:
```
pca_2_data['Cls_label'] = clusters.labels_
pca_2_data['given_cancer_type'] = label.Class.values
pca_2_data.head()
```

[49]:

| | pca1 | pca2 | Cls_label | given_cancer_type |
|---|---|---|---|---|
| 0 | -57.446987 | 95.410981 | 4 | PRAD |
| 1 | -16.919430 | 0.732470 | 0 | LUAD |
| 2 | -70.345218 | -19.303326 | 1 | PRAD |
| 3 | -49.161591 | -9.227586 | 1 | PRAD |
| 4 | -18.132534 | -51.327797 | 2 | BRCA |

[50]:
```
brca = pca_2_data.groupby('given_cancer_type').get_group('BRCA')
brca.Cls_label.value_counts()
```

[50]:
```
2    200
1     42
0     39
4     19
Name: Cls_label, dtype: int64
```

[51]:
```
luad = pca_2_data.groupby('given_cancer_type').get_group('LUAD')
luad.Cls_label.value_counts()
```

[51]:
```
0    88
1    34
2    12
4     6
3     1
Name: Cls_label, dtype: int64
```

[52]:
```
coad = pca_2_data.groupby('given_cancer_type').get_group('COAD')
coad.Cls_label.value_counts()
```

[52]:
```
1    60
4    11
0     7
Name: Cls_label, dtype: int64
```

[53]:
```
prad = pca_2_data.groupby('given_cancer_type').get_group('PRAD')
prad.Cls_label.value_counts()
```

[53]:
```
1    54
0    51
2    16
4    15
Name: Cls_label, dtype: int64
```

[54]:
```
kirc = pca_2_data.groupby('given_cancer_type').get_group('KIRC')
kirc.Cls_label.value_counts()
```

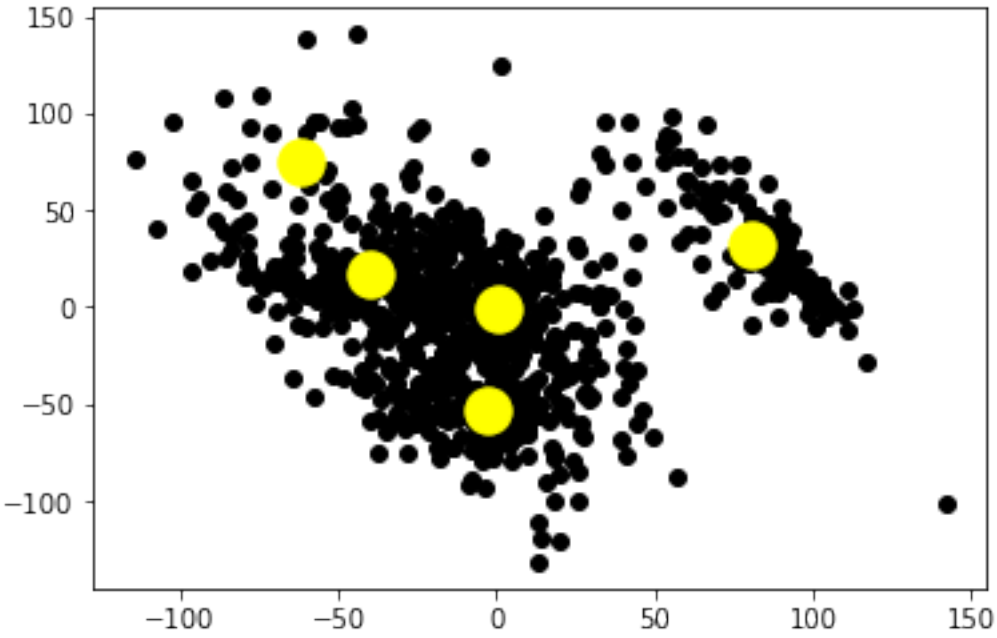
[54]:
```
3    138
0      7
4      1
Name: Cls_label, dtype: int64
```

[55]:
```
clusters.cluster_centers_
```

[55]:
```
array([[  1.44937491,   4.94098556],
       [-41.14524171,  14.02848476],
       [ -2.53772416, -51.80975506],
       [ 81.21177315,  32.92811734],
```

```
       [-60.18008724,  71.44922854]])
```

[56]:
```
kmeans = KMeans(n_clusters=5, init='k-means++', max_iter=3000, n_init=10,␣
 ↪random_state=0)
pred_y = kmeans.fit_predict(X_pca_with_2)
plt.scatter(X_pca_with_2[:,0], X_pca_with_2[:,1],c='black')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],␣
 ↪s=300, c='yellow')
plt.show()
```




**KMEANS Clustering with PCA = .995**

[57]:
```
clu_995 = KMeans(5, n_init = 5)
clu_995.fit(X_pca_with_995)
clu_995.labels_
```

[57]:
```
array([1, 0, 1, 1, 4, 1, 3, 1, 4, 1, 4, 3, 1, 4, 4, 4, 0, 3, 3, 1, 4, 3,
       0, 4, 3, 0, 2, 4, 4, 4, 4, 4, 3, 4, 1, 4, 3, 0, 4, 4, 3, 1, 1, 3,
       3, 4, 1, 2, 4, 0, 4, 0, 4, 1, 2, 4, 4, 2, 3, 4, 0, 4, 4, 0, 1, 2,
       4, 1, 3, 4, 3, 4, 4, 0, 4, 0, 4, 3, 1, 2, 4, 1, 4, 4, 1, 1, 4, 4,
       3, 4, 1, 1, 4, 4, 4, 1, 2, 4, 1, 4, 4, 3, 4, 3, 0, 3, 0, 2, 0, 0,
       1, 4, 0, 1, 4, 3, 3, 3, 4, 4, 0, 3, 0, 4, 1, 1, 1, 4, 3, 4, 2, 4,
       2, 4, 4, 3, 0, 4, 3, 2, 4, 1, 4, 3, 0, 2, 1, 4, 0, 0, 0, 0, 4, 4,
       0, 4, 4, 1, 1, 0, 1, 0, 3, 4, 1, 0, 2, 3, 0, 4, 3, 0, 4, 0, 4, 4,
       4, 1, 4, 3, 2, 3, 4, 1, 1, 1, 0, 0, 4, 0, 0, 3, 0, 1, 0, 4, 4, 4,
```

```
       0, 0, 4, 3, 3, 3, 3, 1, 4, 1, 4, 0, 0, 4, 1, 4, 1, 4, 4, 4, 0, 4,
       3, 0, 3, 3, 0, 4, 3, 1, 4, 0, 0, 1, 2, 4, 3, 1, 3, 2, 4, 3, 3, 0,
       1, 1, 0, 0, 3, 4, 4, 2, 4, 1, 2, 4, 1, 3, 1, 1, 1, 4, 2, 2, 0, 2,
       2, 1, 0, 4, 4, 3, 3, 4, 2, 1, 3, 1, 4, 4, 3, 4, 4, 4, 4, 0, 0, 4,
       4, 4, 3, 3, 3, 3, 4, 4, 4, 3, 4, 4, 0, 1, 4, 4, 2, 0, 1, 4, 4, 4,
       2, 4, 1, 4, 2, 0, 0, 1, 3, 4, 3, 3, 0, 2, 3, 4, 4, 4, 4, 3, 4, 4,
       1, 4, 3, 4, 0, 1, 3, 4, 1, 2, 4, 4, 4, 0, 0, 0, 4, 4, 1, 0, 4, 3,
       4, 2, 2, 0, 4, 3, 4, 4, 4, 2, 0, 2, 3, 1, 3, 4, 4, 3, 4, 2, 1, 0,
       1, 4, 3, 1, 4, 2, 3, 3, 2, 2, 1, 4, 4, 2, 3, 0, 1, 4, 4, 4, 0, 0,
       3, 0, 4, 3, 2, 1, 4, 0, 1, 4, 4, 4, 0, 4, 4, 1, 4, 1, 2, 4, 0, 4,
       4, 0, 4, 4, 4, 3, 0, 1, 4, 1, 3, 4, 3, 2, 4, 1, 0, 3, 4, 4, 3, 4,
       0, 4, 4, 1, 2, 4, 3, 0, 1, 4, 1, 4, 4, 4, 4, 3, 0, 4, 3, 4, 4, 0,
       0, 3, 2, 1, 2, 4, 3, 3, 4, 1, 3, 2, 0, 0, 4, 1, 1, 4, 1, 0, 3, 1,
       4, 0, 1, 0, 4, 4, 2, 0, 3, 2, 0, 4, 1, 4, 4, 1, 4, 2, 4, 2, 3, 4,
       4, 0, 0, 0, 2, 3, 0, 0, 4, 4, 3, 1, 0, 1, 4, 3, 4, 3, 3, 1, 1, 0,
       4, 3, 2, 2, 4, 3, 3, 4, 4, 1, 3, 2, 4, 4, 2, 0, 4, 4, 4, 3, 1, 0,
       0, 4, 3, 2, 3, 3, 4, 1, 0, 3, 4, 2, 0, 0, 0, 1, 0, 3, 4, 4, 2, 1,
       4, 4, 4, 3, 0, 0, 4, 1, 0, 0, 4, 3, 1, 2, 0, 1, 2, 0, 2, 3, 3, 4,
       4, 3, 3, 2, 4, 4, 1, 1, 3, 4, 0, 4, 4, 2, 4, 1, 1, 4, 4, 2, 4, 3,
       4, 4, 2, 4, 1, 4, 4, 3, 1, 0, 4, 4, 3, 4, 4, 4, 4, 4, 2, 0, 0, 4,
       4, 4, 1, 4, 4, 3, 0, 0, 3, 3, 0, 3, 2, 4, 2, 3, 4, 4, 1, 1, 1, 0,
       1, 1, 2, 4, 4, 2, 0, 3, 4, 3, 2, 4, 4, 4, 1, 0, 3, 4, 1, 3, 1, 4,
       0, 3, 1, 0, 1, 1, 4, 3, 1, 0, 2, 2, 4, 4, 4, 0, 3, 3, 3, 4, 0, 3,
       1, 4, 0, 1, 4, 1, 4, 3, 4, 2, 1, 1, 3, 1, 3, 4, 0, 0, 4, 4, 3, 4,
       4, 4, 4, 3, 3, 1, 2, 3, 4, 4, 3, 4, 0, 4, 4, 1, 4, 1, 4, 2, 4, 4,
       3, 0, 4, 4, 1, 4, 1, 2, 4, 4, 4, 0, 4, 0, 4, 0, 3, 3, 2, 2, 4, 4,
       4, 0, 4, 0, 3, 4, 0, 3, 0, 0, 0, 3, 4, 1, 4, 4, 3, 4, 1, 3, 4, 4,
       4, 4, 1, 0, 4, 0, 0, 1, 1], dtype=int32)
```

[58]:
```
pca_995_data = pd.DataFrame(data=X_pca_with_995)
pca_995_data.head()
```

[58]:

| | 0 | 1 | 2 | 3 | 4 | 5 | \ |
|---|---|---|---|---|---|---|---|
| 0 | -62.755415 | -94.071973 | 89.519831 | -15.942567 | 81.423539 | -13.998292 | |
| 1 | -2.432896 | 90.585842 | -1.067308 | -53.083120 | -15.676684 | 60.842472 | |
| 2 | -71.266853 | -8.064608 | 66.112455 | 81.381475 | -7.525685 | 109.824273 | |
| 3 | -84.770785 | -73.244566 | 74.181000 | 27.022697 | -18.044895 | 50.116433 | |
| 4 | -69.560171 | -9.612940 | -67.497549 | 34.868543 | -1.795849 | -6.676780 | |

| | 6 | 7 | 8 | 9 | … | 737 | 738 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 7.716073 | -22.936551 | -32.837892 | -2.202680 | … | 4.970114 | -4.081064 | |
| 1 | 10.257369 | -48.822959 | 14.257400 | -12.214352 | … | -0.477670 | 0.215619 | |
| 2 | 5.519407 | -13.364480 | 38.415728 | -5.124731 | … | 0.263994 | 0.263786 | |
| 3 | -3.495197 | -11.318520 | 8.319656 | -3.149509 | … | -1.037772 | 0.381578 | |
| 4 | -2.840781 | 16.780157 | -49.319753 | 10.508631 | … | 0.533116 | 1.488047 | |

| | 739 | 740 | 741 | 742 | 743 | 744 | 745 | \ |
|---|---|---|---|---|---|---|---|---|

```
0 -0.626193 -1.265756 -0.017984 -2.740860  0.944037  3.092581  0.713598
1 -0.593678 -0.403462  1.181537  0.490910  0.197768  0.013967 -0.395176
2  0.328453  0.004078  0.363928 -1.109210  0.331488  0.128899 -0.264530
3  0.652455 -3.624900 -1.203028 -2.347912  1.577992 -0.781748  0.120442
4  2.767486 -0.631562 -0.794275 -0.514008 -1.875969 -2.526109 -1.073803

        746
0 -0.082122
1 -0.949947
2  0.384594
3 -0.057973
4 -1.161728

[5 rows x 747 columns]
```

[59]:
```
pca_995_data['Cls_label'] = clusters.labels_
pca_995_data['given_cancer_type'] = label.Class.values
```

[60]:
```
pca_995_data.shape
```

[60]:
```
(801, 749)
```

[61]:
```
brca_995 = pca_995_data.groupby('given_cancer_type').get_group('BRCA')
brca_995.Cls_label.value_counts()
```

[61]:
```
2    200
1     42
0     39
4     19
Name: Cls_label, dtype: int64
```

[62]:
```
luad_995 = pca_995_data.groupby('given_cancer_type').get_group('LUAD')
luad_995.Cls_label.value_counts()
```

[62]:
```
0    88
1    34
2    12
4     6
3     1
Name: Cls_label, dtype: int64
```

[63]:
```
coad_995 = pca_995_data.groupby('given_cancer_type').get_group('COAD')
coad_995.Cls_label.value_counts()
```

[63]:
```
1    60
4    11
0     7
```

```
Name: Cls_label, dtype: int64
```

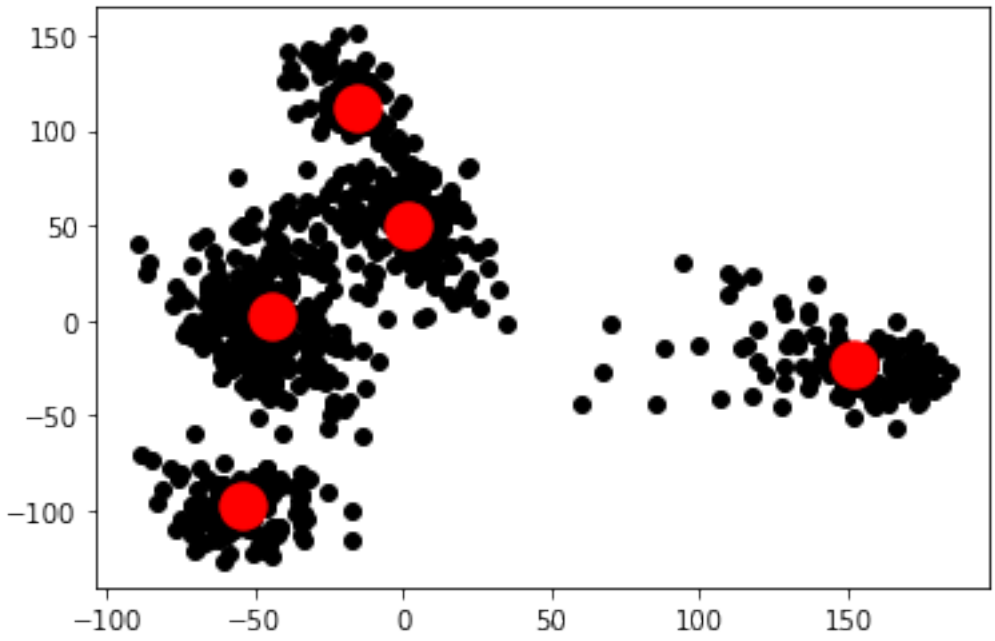
[64]:
```
prad_995 = pca_995_data.groupby('given_cancer_type').get_group('PRAD')
prad_995.Cls_label.value_counts()
```

[64]:
```
1    54
0    51
2    16
4    15
Name: Cls_label, dtype: int64
```

[65]:
```
kirc_995 = pca_995_data.groupby('given_cancer_type').get_group('KIRC')
kirc_995.Cls_label.value_counts()
```

[65]:
```
3    138
0      7
4      1
Name: Cls_label, dtype: int64
```

[66]:
```
kmeans = KMeans(n_clusters=5, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
pred_y = kmeans.fit_predict(X_pca_with_995)
plt.scatter(X_pca_with_995[:,0], X_pca_with_995[:,1],c='black')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],␣
 ↪s=300, c='red')
plt.show()
```



### 0.0.13 Building Classification Model(s) with Feature Selection:

**Build decision tree clasifier**

[67]:
```
ml_x = x_lda
ml_y = y_lda
print(ml_x.shape)
print(ml_y.shape)
```

```
(801, 20531)
(801,)
```

[68]:
```
x_train, x_test, y_train, y_test = train_test_split(ml_x,ml_y,test_size=0.
↪30,random_state=30)
```

[69]:
```
dt_clf = tree.DecisionTreeClassifier(max_depth=5)
dt_clf.fit(x_train,y_train)
dt_clf.score(x_test,y_test)
print('accuracy = ',dt_clf.score(x_test,y_test))
```

```
accuracy =  0.9585062240663901
```

**SVM**

[70]:
```
sv_clf = SVC(probability=True, kernel='linear')
sv_clf.fit(x_train,y_train)
sv_clf.score(x_test,y_test)
y_pred = sv_clf.predict(x_test)
print('accuracy = ',accuracy_score (y_test,y_pred))
```

```
accuracy =  1.0
```

**Random Forest**

[71]:
```
rf_clf = ensemble.RandomForestClassifier(n_estimators=100)
rf_clf.fit(x_train,y_train)
print('accuracy = ',rf_clf.score(x_test,y_test))
```

```
accuracy =  0.9875518672199171
```

**Naive Bayes Classifier**

**Bayes Theorem:**

[72]:
```
gb_clf = GaussianNB()
gb_clf.fit(x_train,y_train)
```

```
print('accuracy = ',gb_clf.score(x_test,y_test))
```

```
accuracy =  0.7385892116182573
```

**KNN Classifier**

[73]:
```
knn_clf = KNeighborsClassifier(n_neighbors=5)
knn_clf.fit(x_train,y_train)
print('accuracy = ',knn_clf.score(x_test,y_test))
```

```
accuracy =  0.995850622406639
```

### 0.0.14 One way F test

[74]:
```
df_tsne = pd.DataFrame(data=tnse_features,columns=['tsne1','tsne2'])
df_tsne['cancer_type']=label['Class']
df_tsne
```

[74]:

| | tsne1 | tsne2 | cancer_type |
|---|---|---|---|
| 0 | 32.087540 | 4.844350 | PRAD |
| 1 | -6.368939 | 13.379185 | LUAD |
| 2 | 29.596621 | 16.001747 | PRAD |
| 3 | 29.515751 | 15.109866 | PRAD |
| 4 | 4.107100 | -22.900890 | BRCA |
| .. | … | … | … |
| 796 | 3.455332 | -23.795534 | BRCA |
| 797 | -3.618943 | 22.222263 | LUAD |
| 798 | -7.533560 | 38.971920 | COAD |
| 799 | 27.988205 | 9.964670 | PRAD |
| 800 | 28.788488 | 9.019204 | PRAD |

[801 rows x 3 columns]

[75]:
```
df_anova_tsne = df_tsne[['tsne2','cancer_type']]
grps_tsne = pd.unique(df_anova_tsne.cancer_type.values)

d_data = {grp:df_anova_tsne['tsne2'][df_anova_tsne.cancer_type == grp] for grp␣
 ↪in grps_tsne}

F, p = stats.f_oneway(d_data['LUAD'], d_data['PRAD'], d_data['BRCA'],␣
 ↪d_data['KIRC'], d_data['COAD'])

if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
reject null hypothesis
```

[76]:
```
df_anova_tsne = df_tsne[['tsne1','cancer_type']]
grps_tsne = pd.unique(df_anova_tsne.cancer_type.values)

d_data = {grp:df_anova_tsne['tsne1'][df_anova_tsne.cancer_type == grp] for grp␣
 ↪in grps_tsne}

F, p = stats.f_oneway(d_data['LUAD'], d_data['PRAD'], d_data['BRCA'],␣
 ↪d_data['KIRC'], d_data['COAD'])

if p<0.05:
    print("reject null hypothesis")
else:
    print("accept null hypothesis")
```

```
reject null hypothesis
```

### 0.0.15 DNN

[77]:
```
features=merged_data.drop(['Unnamed: 0'],axis=1)
features=features.drop(['Class'],axis=1)
target=merged_data['Class']
```

[78]:
```
features.head()
```

[78]:

| | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | gene_6 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 2.017209 | 3.265527 | 5.478487 | 10.431999 | 0.0 | 7.175175 | |
| 1 | 0.0 | 0.592732 | 1.588421 | 7.586157 | 9.623011 | 0.0 | 6.816049 | |
| 2 | 0.0 | 3.511759 | 4.327199 | 6.881787 | 9.870730 | 0.0 | 6.972130 | |
| 3 | 0.0 | 3.663618 | 4.507649 | 6.659068 | 10.196184 | 0.0 | 7.843375 | |
| 4 | 0.0 | 2.655741 | 2.821547 | 6.539454 | 9.738265 | 0.0 | 6.566967 | |

| | gene_7 | gene_8 | gene_9 | … | gene_20521 | gene_20522 | gene_20523 | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.591871 | 0.0 | 0.0 | … | 4.926711 | 8.210257 | 9.723516 | |
| 1 | 0.000000 | 0.0 | 0.0 | … | 4.593372 | 7.323865 | 9.740931 | |
| 2 | 0.452595 | 0.0 | 0.0 | … | 5.125213 | 8.127123 | 10.908640 | |
| 3 | 0.434882 | 0.0 | 0.0 | … | 6.076566 | 8.792959 | 10.141520 | |
| 4 | 0.360982 | 0.0 | 0.0 | … | 5.996032 | 8.891425 | 10.373790 | |

| | gene_20524 | gene_20525 | gene_20526 | gene_20527 | gene_20528 | gene_20529 | \ |
|---|---|---|---|---|---|---|---|
| 0 | 7.220030 | 9.119813 | 12.003135 | 9.650743 | 8.921326 | 5.286759 | |
| 1 | 6.256586 | 8.381612 | 12.674552 | 10.517059 | 9.397854 | 2.094168 | |
| 2 | 5.401607 | 9.911597 | 9.045255 | 9.788359 | 10.090470 | 1.683023 | |
| 3 | 8.942805 | 9.601208 | 11.392682 | 9.694814 | 9.684365 | 3.292001 | |
| 4 | 7.181162 | 9.846910 | 11.922439 | 9.217749 | 9.461191 | 5.110372 | |

```
   gene_20530
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0

[5 rows x 20531 columns]
```

[79]:
```
target.head()
```

[79]:
```
0    1
1    2
2    1
3    1
4    3
Name: Class, dtype: int64
```

[80]:
```
f_1 =features.values
```

[81]:
```
y_1 = pd.get_dummies(y_lda)
```

[82]:
```
from sklearn.model_selection import train_test_split

X1_train, X1_valid, y1_train, y1_valid = train_test_split(f_1,y_1, test_size =␣
↪0.30, random_state=42)
```

[83]:
```
print(X1_train.shape)
print(X1_valid.shape)
print(y1_valid.shape)
print(y1_train.shape)
```

```
(560, 20531)
(241, 20531)
(241, 5)
(560, 5)
```

### 0.0.16 Define the model

**Optimizer is chosen SGD**

[84]:
```
#Initialize
model = tf.keras.models.Sequential()

#adding layers
```

```
model.add(tf.keras.layers.Dense(10000, input_dim=20531, activation='relu',␣
 ↪kernel_initializer='he_uniform'))

#Normalize the data
model.add(tf.keras.layers.BatchNormalization())

#Adding hidden layer
model.add(tf.keras.layers.Dense(5000, activation='relu'))

model.add(tf.keras.layers.Dense(2000, activation='relu'))

model.add(tf.keras.layers.Dense(1000, activation='relu'))

model.add(tf.keras.layers.Dense(500, activation='relu'))

model.add(tf.keras.layers.Dense(200, activation='relu'))

model.add(tf.keras.layers.Dense(100, activation='relu'))

#Add OUTPUT layer
model.add(tf.keras.layers.Dense(5, activation='softmax'))

#Create optimizer with non-default learning rate
sgd_optimizer = tf.keras.optimizers.SGD(learning_rate=0.01)

#Compile the model
model.compile(optimizer=sgd_optimizer, loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
```

[85]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10000)             205320000

 batch_normalization (BatchN  (None, 10000)            40000
 ormalization)

 dense_1 (Dense)             (None, 5000)              50005000

 dense_2 (Dense)             (None, 2000)              10002000

 dense_3 (Dense)             (None, 1000)              2001000

 dense_4 (Dense)             (None, 500)               500500
```

```
 dense_5 (Dense)             (None, 200)               100200

 dense_6 (Dense)             (None, 100)               20100

 dense_7 (Dense)             (None, 5)                 505

=================================================================
Total params: 267,989,305
Trainable params: 267,969,305
Non-trainable params: 20,000
_________________________________________________________________
```

[86]:
```
epoch = model.fit(X1_train,y1_train,
          validation_data=(X1_valid,y1_valid),
          epochs=5,
          batch_size=32)
```

```
Epoch 1/5
18/18 [==============================] - 28s 2s/step - loss: 0.6122 - accuracy:
0.8786 - val_loss: 1.8243 - val_accuracy: 0.6929
Epoch 2/5
18/18 [==============================] - 28s 2s/step - loss: 0.0651 - accuracy:
0.9982 - val_loss: 1.0982 - val_accuracy: 0.7967
Epoch 3/5
18/18 [==============================] - 28s 2s/step - loss: 0.0257 - accuracy:
1.0000 - val_loss: 0.1970 - val_accuracy: 0.9004
Epoch 4/5
18/18 [==============================] - 28s 2s/step - loss: 0.0131 - accuracy:
1.0000 - val_loss: 0.1351 - val_accuracy: 0.9502
Epoch 5/5
18/18 [==============================] - 28s 2s/step - loss: 0.0093 - accuracy:
1.0000 - val_loss: 0.0700 - val_accuracy: 0.9876
```

[87]:
```
abc = model.predict(X1_valid)
```

[88]:
```
y_pre=[]
for k in abc:
    y_pre.append(np.argmax(k))

y_val=[]
for k in y1_valid.values:
    y_val.append(np.argmax(k))
```

[89]:
```
# Making the Confusion Matrix
confusion_matrix(y_val, y_pre)
```

[89]:
```
array([[42,  0,  0,  0,  0],
       [ 0, 40,  0,  0,  0],
       [ 0,  0, 90,  0,  0],
       [ 0,  0,  0, 41,  0],
       [ 0,  3,  0,  0, 25]])
```

### Evaluate the model

[90]:
```
train_acc = model.evaluate(X1_train, y1_train, verbose=0)
test_acc = model.evaluate(X1_valid, y1_valid, verbose=0)
```

[91]:
```
print(" Train Accuracy = ",train_acc[1])
```

```
 Train Accuracy =  0.9714285731315613
```

[92]:
```
print(" Test Accuracy = ",test_acc[1])
```

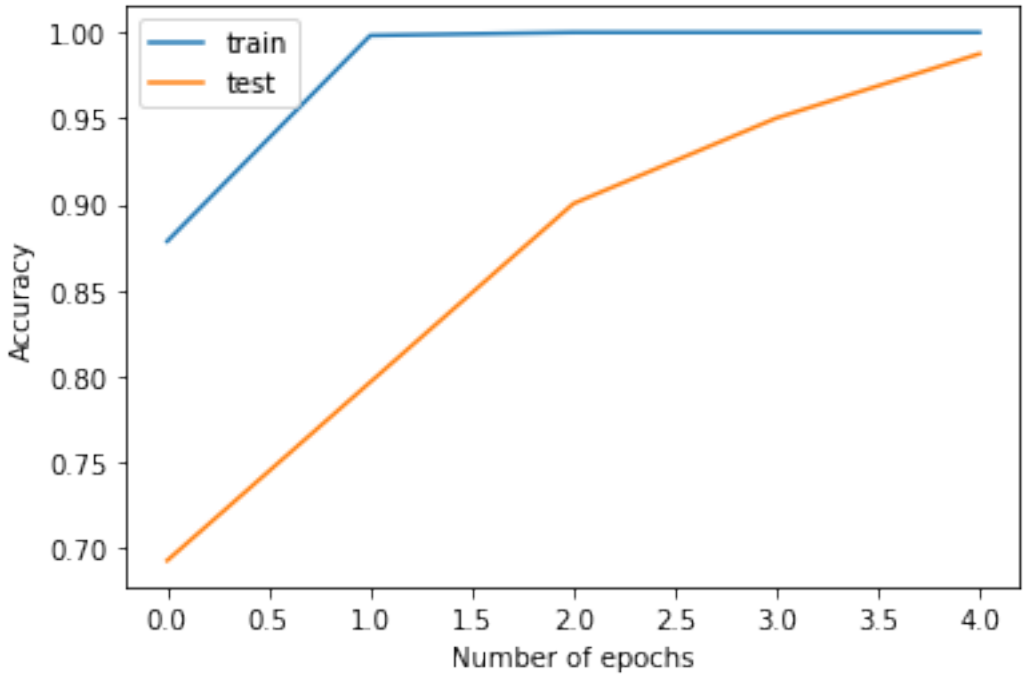
```
 Test Accuracy =  0.9875518679618835
```

[93]:
```
print(classification_report(y_val,y_pre))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        42
           1       0.93      1.00      0.96        40
           2       1.00      1.00      1.00        90
           3       1.00      1.00      1.00        41
           4       1.00      0.89      0.94        28

    accuracy                           0.99       241
   macro avg       0.99      0.98      0.98       241
weighted avg       0.99      0.99      0.99       241
```

### Plot History

[94]:
```
plt.plot(epoch.history['accuracy'], label='train')
plt.plot(epoch.history['val_accuracy'], label='test')
plt.xlabel('Number of epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```


train
test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Accuracy
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of epochs