ITXXX-LAB_ASSIGNMENT1

Akshara

211AI012

Dataset:

WHO life_expectancy data was chosen the dataset consists of 22 columns. I chose 'status' column as label (Developed or developing). This dataset consists of data(like life expectancy,schooling,gdp,per capita income etc) of 193 countries over the period of 2000 to 2015. Using status column as label and the rest 21 as features we can transform this into a binary classification problem where we can classify whether a country is developed or developing for the upcoming years using its features data.

Link: https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

Google colab notebook:

code can be found here –

https://colab.research.google.com/drive/1mq9E1rm8wf5eY-kIjpIvpsaHSy4Du9mT?usp=sharing

Tasks:

A. Visualize the data and explain why PCA or SVD must be used

In the dataset taken there are 21 features. The training samples size is of only 2350 rows this may lead to the model to overfit (curse of dimensionality). moreover there can be redundant features which depend on each other. This will lead to a poor model.

1.

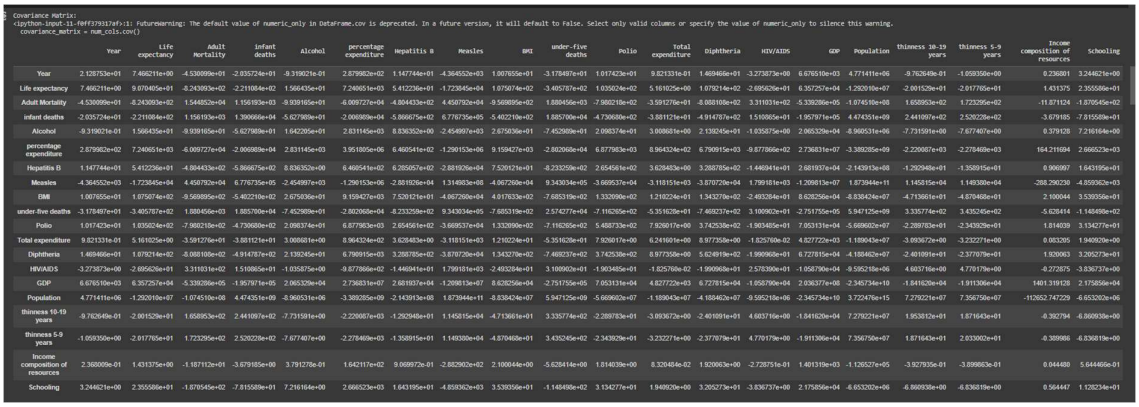
```
<ipython-input-10-c1b0e571dc2f>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  num_cols.corr()
```

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 10-19 years	thinness 5-9 years	Income composition of resources	Schooling
Year	1.000000	0.170033	-0.079052	-0.037415	-0.052990	0.031400	0.104333	-0.082493	0.108974	-0.042937	0.094158	0.090740	0.134337	-0.139741	0.101620	0.016969	-0.047876	-0.050929	0.243468	0.209400
Life expectancy	0.170033	1.000000	-0.696359	-0.196557	0.404877	0.381864	0.256762	-0.157586	0.567694	-0.222529	0.465556	0.218086	0.479495	-0.556556	0.461455	-0.021538	-0.477183	-0.471584	0.724776	0.751975
Adult Mortality	-0.079052	-0.696359	1.000000	0.078756	-0.195848	-0.242860	-0.162476	0.031176	-0.387017	0.094146	-0.274823	-0.115281	-0.275131	0.523821	-0.296049	-0.013647	0.302904	0.308457	-0.457626	-0.454612
infant deaths	-0.037415	-0.196557	0.078756	1.000000	-0.115638	-0.085612	-0.223566	0.501128	-0.227279	0.996629	-0.170689	-0.128616	-0.175171	0.025231	-0.108427	0.556801	0.465711	0.471350	-0.145139	-0.193720
Alcohol	-0.052990	0.404877	-0.195848	-0.115638	1.000000	0.341285	0.087549	-0.051827	0.330408	-0.112370	0.221734	0.296942	0.222029	-0.048845	0.354712	-0.035252	-0.428795	-0.417414	0.450040	0.547378
percentage expenditure	0.031400	0.381864	-0.242860	-0.085612	0.341285	1.000000	0.016274	-0.056596	0.228700	-0.087852	0.147259	0.174420	0.143624	-0.097857	0.899373	-0.025662	-0.251369	-0.252905	0.381952	0.389687
Hepatitis B	0.104333	0.256762	-0.162476	-0.223566	0.087549	0.016274	1.000000	-0.120529	0.150380	-0.233126	0.486171	0.058280	0.611495	-0.112675	0.083903	-0.123321	-0.120429	-0.124960	0.199549	0.231117
Measles	-0.082493	-0.157586	0.031176	0.501128	-0.051827	-0.056596	-0.120529	1.000000	-0.175977	0.507809	-0.136166	-0.106241	-0.141882	0.030899	-0.076466	0.265966	0.224808	0.221672	-0.129568	-0.137225
BMI	0.108974	0.567694	-0.387017	-0.227279	0.330408	0.228700	0.150380	-0.175977	1.000000	-0.237669	0.284569	0.242503	0.283147	-0.243717	0.301557	-0.072301	-0.532025	-0.538911	0.568774	0.546961
under-five deaths	-0.042937	-0.222529	0.094146	0.996629	-0.112370	-0.087852	-0.233126	0.507809	-0.237669	1.000000	-0.188720	-0.130148	-0.195668	0.038062	-0.112681	0.544423	0.467789	0.472263	-0.163305	-0.209373
Polio	0.094158	0.465556	-0.274823	-0.170689	0.221734	0.147259	0.486171	-0.136166	0.284569	-0.188720	1.000000	0.137330	0.673553	-0.159560	0.211976	-0.038540	-0.221823	-0.222582	0.381078	0.417866
Total expenditure	0.090740	0.218086	-0.115281	-0.128616	0.296942	0.174420	0.058280	-0.106241	0.242503	-0.130148	0.137330	1.000000	0.152754	-0.001389	0.138364	-0.079662	-0.277101	-0.283774	0.166682	0.246384
Diphtheria	0.134337	0.479495	-0.275131	-0.175171	0.222029	0.143624	0.611495	-0.141882	0.283147	-0.195668	0.673553	0.152754	1.000000	-0.164860	0.209666	-0.028444	-0.229518	-0.222743	0.401456	0.425332
HIV/AIDS	-0.139741	-0.556556	0.523821	0.025231	-0.048845	-0.097857	-0.112675	0.030899	-0.243717	0.038062	-0.159560	-0.001389	-0.164860	1.000000	-0.136491	-0.027854	0.204064	0.207283	-0.249519	-0.220429
GDP	0.101620	0.461455	-0.296049	-0.108427	0.354712	0.899373	0.083903	-0.076466	0.301557	-0.112681	0.211976	0.138364	0.209666	-0.136491	1.000000	-0.028270	-0.285697	-0.290539	0.460341	0.448273
Population	0.016969	-0.021538	-0.013647	0.556801	-0.035252	-0.025662	-0.123321	0.265966	-0.072301	0.544423	-0.038540	-0.079662	-0.028444	-0.027854	-0.028270	1.000000	0.253944	0.251403	-0.008735	-0.031668
thinness 10-19 years	-0.047876	-0.477183	0.302904	0.465711	-0.428795	-0.251369	-0.120429	0.224808	-0.532025	0.467789	-0.221823	-0.277101	-0.229518	0.204064	-0.285697	0.253944	1.000000	0.939102	-0.422429	-0.471652
thinness 5-9 years	-0.050929	-0.471584	0.308457	0.471350	-0.417414	-0.252905	-0.124960	0.221672	-0.538911	0.472263	-0.222582	-0.283774	-0.222743	0.207283	-0.290539	0.251403	0.939102	1.000000	-0.411053	-0.460632
Income composition of resources	0.243468	0.724776	-0.457626	-0.145139	0.450040	0.381952	0.199549	-0.129568	0.568774	-0.163305	0.381078	0.166682	0.401456	-0.249519	0.460341	-0.008735	-0.422429	-0.411053	1.000000	0.800092
Schooling	0.209400	0.751975	-0.454612	-0.193720	0.547378	0.389687	0.231117	-0.137225	0.546961	-0.209373	0.417866	0.246384	0.425332	-0.220429	0.448273	-0.031668	-0.471652	-0.460632	0.800092	1.000000

This is the correlation matrix of the data we can see that some features are highly correlated with each other. Examples : schooling and life expectancy(0.75), infant deaths and underfive deaths(0.99),polio and diphtheria(0.67),income composition and schooling(0.8) etc. there are a lot of features that have correlation >0.4 or <-0.4 between them. This explains that we have remove correlation in features in our data. So we use SVD or PCA. SVD or PCA can transform the original features into a new set of uncorrelated variables, known as principal components. This can be beneficial because

it **reduces the impact of multicollinearity**, making the model more robust and interpretable.

2.



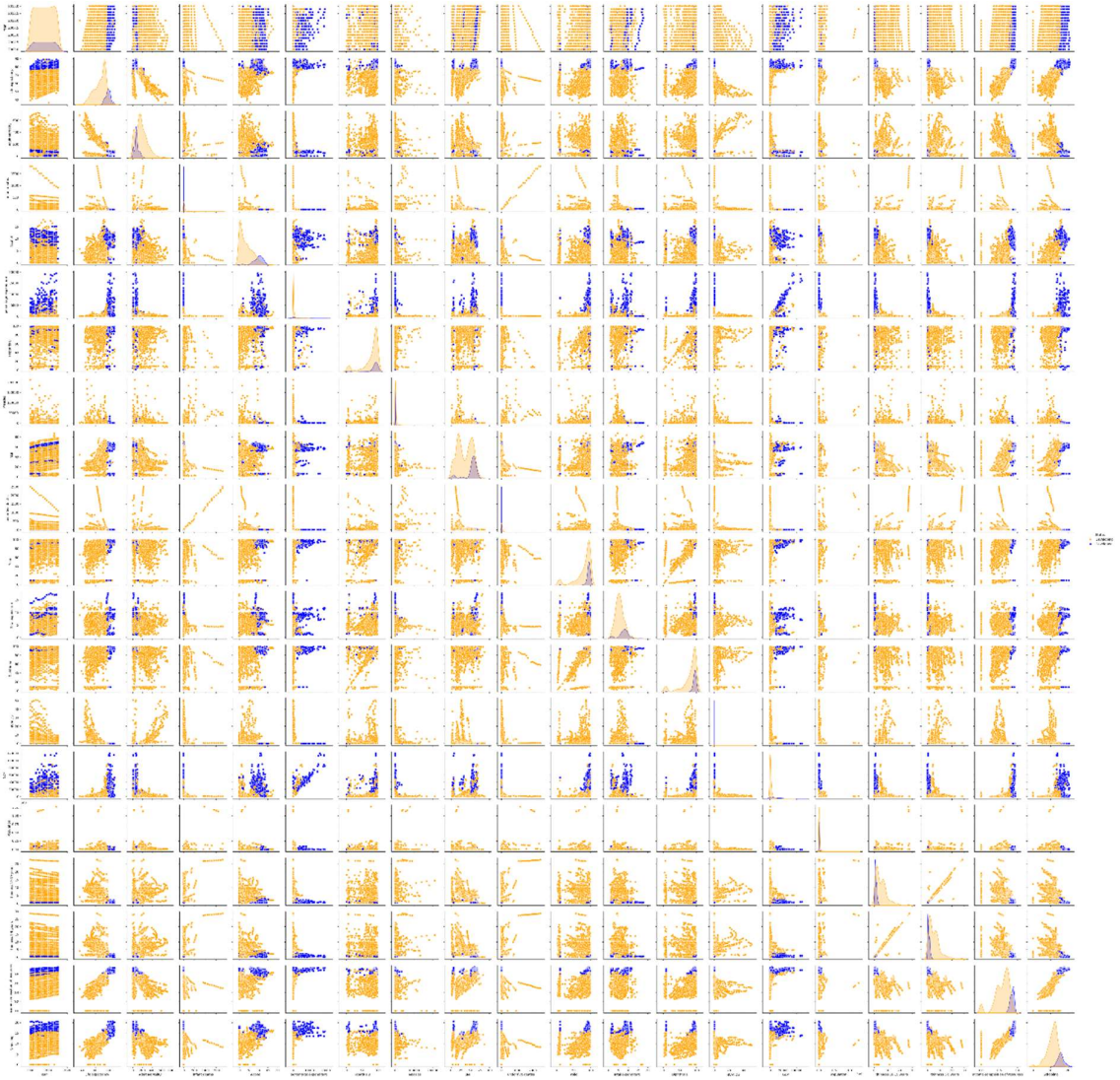
Covariance Matrix:
<ipython-input-11-f6ff379317af>:1: FutureWarning: The default value of numeric_only in DataFrame.cov is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
covariance_matrix = num_cols.cov()

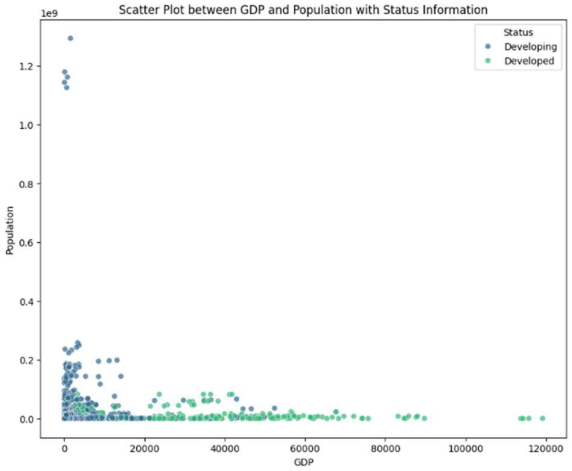
	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 10-19 years	thinness 5-9 years	Income composition of resources	Schooling
Year	2.128753e+01	7.468211e+00	-4.530999e+01	-2.035724e+01	-9.319021e-01	2.879982e+02	1.147744e+01	-4.364552e+03	1.007655e+01	-3.178497e+01	1.017423e+01	9.821331e-01	1.469466e+01	-3.273873e+00	6.676510e+03	4.771411e+06	-9.762648e-01	-1.059350e+00	0.236801	3.244621e+00
Life expectancy	7.468211e+00	9.070405e+01	-8.243093e+02	-2.211084e+02	1.566435e+01	7.240651e+03	5.412236e+01	-1.723845e+04	1.075074e+02	-3.405787e+02	1.035024e+02	5.161025e+00	1.079214e+02	-2.695626e+01	6.357257e+04	-1.292010e+07	-2.001529e+01	-2.017765e+01	1.431375	2.355586e+01
Adult Mortality	-4.530999e+01	-8.243093e+02	1.544852e+04	1.156193e+03	-9.939165e+01	-6.009727e+04	-4.804433e+02	4.450792e+04	-9.569895e+02	1.880456e+03	-7.980218e+02	-3.591276e+01	-8.081108e+02	3.311031e+02	-5.338286e+05	-1.074510e+08	1.658953e+02	1.723295e+02	-11.871124	-1.870545e+02
infant deaths	-2.035724e+01	-2.211084e+02	1.156193e+03	1.300666e+04	-5.627989e+01	-2.006989e+04	-5.866675e+02	6.776735e+05	-5.402210e+02	1.885700e+04	-4.730690e+02	-3.881121e+01	-4.914787e+02	1.510865e+01	-1.957971e+05	4.474351e+09	2.441097e+02	2.520228e+02	-3.679105	-7.815589e+01
Alcohol	-9.319021e-01	1.566435e+01	-9.939165e+01	-5.627989e+01	1.642205e+01	2.831145e+03	8.836352e+00	-2.454997e+03	2.675036e+01	-7.452989e+01	2.098374e+01	3.008681e+00	2.139245e+01	-1.035875e+00	2.065329e+04	-8.960531e+06	-7.731591e+00	-7.677407e+00	0.379128	7.216164e+00
percentage expenditure	2.879982e+02	7.240651e+03	-6.009727e+04	-2.006989e+04	2.831145e+03	3.951085e+06	6.460541e+02	-1.290153e+06	9.159427e+03	-2.862068e+04	6.877983e+03	8.964324e+02	6.790915e+03	-9.877866e+02	2.736831e+07	-3.389285e+09	-2.220067e+03	-2.278468e+03	164.211694	2.666523e+03
Hepatitis B	1.147744e+01	5.412236e+01	-4.804433e+02	-5.866675e+02	8.836352e+00	6.460541e+02	6.285057e+02	-2.881926e+04	7.520121e+01	-8.233259e+02	2.654561e+02	3.628483e+00	3.288785e+02	-1.446941e+01	2.681937e+04	-2.143913e+08	-1.292948e+01	-1.358915e+01	0.906997	1.643195e+01
Measles	-4.364552e+03	-1.723845e+04	4.450792e+04	6.776735e+05	-2.454997e+03	-1.290153e+06	-2.881926e+04	1.314983e+08	-4.067260e+04	9.343034e+05	-3.669537e+04	-3.118151e+03	-3.870720e+04	1.799181e+03	-1.209813e+07	1.873944e+11	1.145815e+04	1.149388e+04	-288.290238	-4.859362e+03
BMI	1.007655e+01	1.075074e+02	-9.569895e+02	-5.402210e+02	2.675036e+01	9.159427e+03	7.520121e+01	-4.067260e+04	4.017633e+02	-7.685319e+02	1.332090e+02	1.210224e+01	1.343279e+02	-2.493204e+01	8.628256e+04	-8.838424e+07	-4.713661e+01	-4.870468e+01	2.100044	3.539556e+01
under-five deaths	-3.178497e+01	-3.405787e+02	1.880456e+03	1.885700e+04	-7.452989e+01	-2.802068e+04	-8.233259e+02	9.343034e+05	-7.685319e+02	2.574277e+04	-7.116285e+02	-5.351628e+01	-7.469237e+02	3.100902e+01	-2.751755e+05	5.947125e+09	3.335774e+02	3.435245e+02	-5.628414	-1.148498e+02
Polio	1.017423e+01	1.035024e+02	-7.980218e+02	-4.730690e+02	2.098374e+01	6.877983e+03	2.654561e+02	-3.669537e+04	1.332090e+02	-7.116285e+02	5.488733e+02	7.926017e+00	3.742538e+02	-1.903485e+01	7.053131e+04	-5.669602e+07	-2.289783e+01	-2.343929e+01	1.814039	3.134277e+01
Total expenditure	9.821331e-01	5.161025e+00	-3.591276e+01	-3.881121e+01	3.008681e+00	8.964324e+02	3.628483e+00	-3.118151e+03	1.210224e+01	-5.351628e+01	7.926017e+00	6.241601e+00	8.977358e+00	-1.825760e-02	4.827722e+03	-1.189043e+07	-3.093672e+00	-3.261991e+00	0.083205	1.940920e+00
Diphtheria	1.469466e+01	1.079214e+02	-8.081108e+02	-4.914787e+02	2.139245e+01	6.790915e+03	3.288785e+02	-3.870720e+04	1.343278e+02	-7.469237e+02	3.742538e+02	8.977358e+00	5.624919e+02	-1.990968e+01	6.727815e+04	-4.188462e+07	-2.401091e+01	-2.377079e+01	1.920063	3.205273e+01
HIV/AIDS	-3.273873e+00	-2.695626e+01	3.311031e+02	1.510865e+01	-1.035875e+00	-9.877866e+02	-1.446941e+01	1.799181e+03	-2.493204e+01	3.100902e+01	-1.903485e+01	-1.825760e-02	-1.990968e+01	2.578390e+01	-1.058790e+04	-9.595218e+06	4.603716e+00	4.770179e+00	-0.272875	-3.836737e+00
GDP	6.676510e+03	6.357257e+04	-5.338286e+05	-1.957971e+05	2.065329e+04	2.736831e+07	2.681937e+04	-1.209813e+07	8.628256e+04	-2.751755e+05	7.053131e+04	4.827722e+03	6.727815e+04	-1.058790e+04	2.036377e+08	-2.345734e+10	3.722476e+15	7.279221e+07	7.356750e+07	1401.319128	2.175856e+04
Population	4.771411e+06	-1.292010e+07	-1.074510e+08	4.474351e+09	-8.960531e+06	-3.389285e+09	-2.143913e+08	1.873944e+11	-8.838424e+07	5.947125e+09	-5.669602e+07	-1.189043e+07	-4.188462e+07	-9.595218e+06	-2.345734e+10	3.722476e+15	7.279221e+07	7.356750e+07	-112652.747229	-6.653202e+06
thinness 10-19 years	-9.762649e-01	-2.001529e+01	1.658953e+02	2.441097e+02	-7.731591e+00	-2.220087e+03	-1.292948e+01	1.145815e+04	-4.713661e+01	3.335774e+02	-2.289783e+01	-3.093672e+00	-2.401091e+01	4.603716e+00	-1.041620e+04	7.279221e+07	1.953812e+01	1.871643e+01	-0.392794	-6.860938e+00
thinness 5-9 years	-1.059350e+00	-2.017765e+01	1.723295e+02	2.520228e+02	-7.677407e+00	-2.278469e+03	-1.358915e+01	1.149388e+04	-4.870468e+01	3.435245e+02	-2.343929e+01	-3.232271e+00	-2.377079e+01	4.770179e+00	-1.911306e+04	7.356750e+07	1.871643e+01	2.033002e+01	-0.389986	-6.836819e+00
Income composition of resources	2.368009e-01	1.431375e+00	-1.187112e+01	-3.679185e+00	3.791278e-01	1.642117e+02	9.068972e-01	-2.882902e+02	2.100044e+00	-5.628414e+00	1.814039e+00	8.320484e-02	1.920063e+00	-2.728751e-01	1.401319e+03	-1.126527e+05	-3.927935e-01	-3.899863e-01	0.044488	5.644466e-01
Schooling	3.244621e+00	2.355586e+01	-1.870545e+02	-7.815589e+01	7.216164e+00	2.666523e+03	1.643195e+01	-4.859362e+03	3.539356e+01	-1.148498e+02	3.134277e+01	1.940920e+00	3.205273e+01	-3.836737e+00	2.175856e+04	-6.653202e+06	-6.860938e+00	-6.836819e+00	0.564447	1.128234e+01

This is a covariance matrix. This also demonstrates that some features like life expectancy and income composition, infant deaths and under five deaths are covariant. So we need to reduce redundancy and dimensionality of the data for which PCA or SVD can be used. SVD and PCA can help us identify and retain the most important features, **reducing the dimensionality** while preserving most of the variance in the data. This can help remove noise and focus on the essential information.

3.

As we can see below we have scatter pair plot of every column with every other column (zoom in for better view).the blue dots represent developed and orange samples represent developing. We can see some features have linear(or nearly linear) plots like infant deaths and under five deaths, gdp and percentage expenditure, Thinness(5-9) and thinness(10-19). This demonstrates the collinearity in our data. PCA can help address collinearity issues by transforming them into uncorrelated principal components. Moreover Reducing the dimensionality of data a few principal components can facilitate better visualization, allowing us to explore and understand the structure of thedata. This can be particularly helpful when dealing with high-dimensional datasets.




Scatter Plot between GDP and Population with Status Information
1e9
Status
Developing
Developed
Population
GDP
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20000
40000
60000
80000
100000
120000

B. Implement SVD and PCA logic on your own and find the appropriate k-dimensions for the data

1. **SVD**
    a. The svd_decomposition function takes the matrix X_std and performs Singular Value Decomposition (SVD).The result includes left singular vectors (U), singular values (Sigma), and right singular vectors (Vt).This function is implemented using block power iteration method. Block Power Iteration iteratively refines an initial approximation to the left singular vectors of the matrix through the computation of matrix products and QR decompositions. It is particularly useful when dealing with large matrices where computing the full SVD might be computationally expensive.
    b. Then we decide k based on desired variance to retain. We get k=11 for 90% variance. we calculate cumulative explained variance from the singular values and determines the number of components (k) needed to retain a specified amount of variance (desired_variance).
    c. We retain only the top k components of the SVD results: left singular vectors (U_k), a diagonal matrix of top k singular values (Sigma_k), and the top k right singular vectors (Vt_k).
    d. It reconstructs the data using the reduced SVD components: U_k, Sigma_k, and Vt_k. we can do this either by U_k.dot(Sigma_k)or (X_std).dot(Vt_k.T)

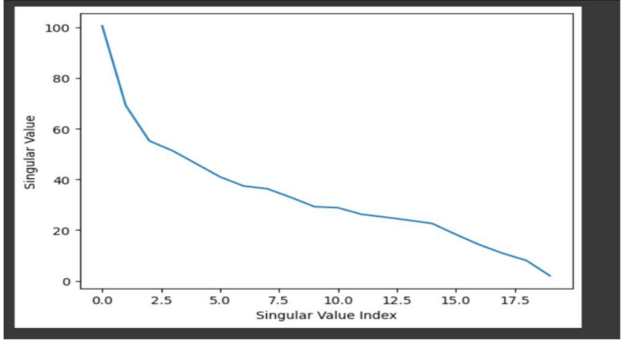
    Output:

```
Number of components to retain 90.0% variance: 11
shape of reduced U matrix (1649, 11)
shape of reduced Singular matrix (11, 11)
shape of reduced V matrix (11, 20)
shape of reduced reconstructed matrix (1649, 11)

] print(X_reconstructed)

[[ 3.88661133 -0.21180361 -0.94791183 ...  0.95813846  0.27348378
   1.79095611]
 [ 3.72185153  0.04760947 -0.05952145 ...  1.02693597  0.42640473
   0.321598  ]
 [ 3.77547659 -0.14967875  0.13584362 ...  0.74682281  0.37743529
   0.23883825]
 ...
 [ 2.30520554  2.36755519 -1.5285449  ... -0.3818512   1.55339677
  -0.6890041 ]
 [ 3.60996401  3.38076397 -1.57050041 ... -0.97242557  1.38470432
  -0.59258821]
 [ 4.69144366  2.82302625 -0.7520181  ... -0.18455246  2.36652763
  -0.23282768]]
```

```
plt.plot(Sigma)
plt.xlabel('Singular Value Index')
plt.ylabel('Singular Value')
plt.show()
```



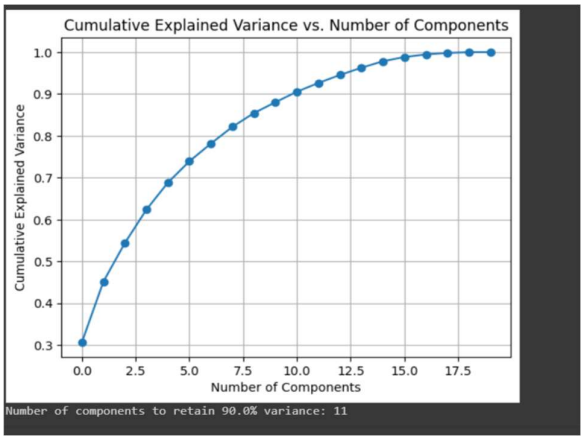
When we plot singular values we see that only about 10 or so singular value indexes have significantly high singular values the rest have very low. This demonstrates that we

can summarise most of the variance in our data by choosing an appropriate k (here we chose 11)

2. **PCA**

   a. First we standardise the data and compute the covariance matrix from the standardized data. The covariance matrix represents the relationships between different features in the dataset
   b. Then we compute the eigenvalues and corresponding eigenvectors of the covariance matrix. Eigenvectors represent the directions of maximum variance, and eigenvalues indicate the magnitude of the variance in those directions.
   c. We sort the eigenvalues in descending order and rearranges the corresponding eigenvectors accordingly. This step is crucial as it allows identifying the principal components in order of significance.
   d. Next we the cumulative explained variance, which represents the proportion of total variance explained by each principal component and the sum of all previous components.
   e. We plot the cumulative explained variance against the number of components. This visualization helps in determining how many components are needed to retain a certain percentage of the total variance.
   f. We Set a threshold (desired_variance=90%) for the cumulative explained variance and determines the minimum number of components (k) needed to achieve at least that much variance.
   g. We get k=11
   h. Then we chose top 11 eigenvectors based on their eigen values and project it onto initial X_Std data to get the final transformed data

   Output:



```
Dimensions of standardized_data: (1649, 20)
Dimensions of top_eigenvectors: (20, 11)
Dimensions of transformed_data: (1649, 11)
```

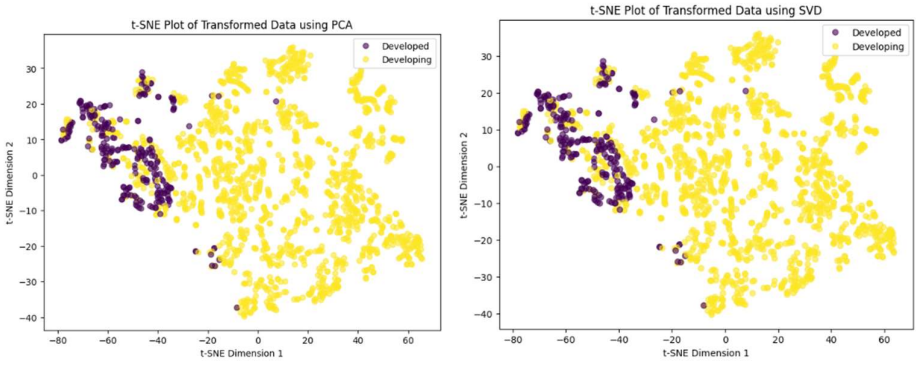
```
transformed_data
```

```
array([[-3.88661133,  0.21180361, -0.9478715 , ...,  0.95814995,
        -0.30265613,  1.7862947 ],
       [-3.72185153, -0.04760947, -0.05959068, ...,  1.02694563,
         -0.4315925 ,  0.31454981],
       [-3.77547659,  0.14967875,  0.13575916, ...,  0.74683278,
         -0.38127993,  0.23260972],
       ...,
       [-2.30520554, -2.36755519, -1.52908612, ..., -0.38185435,
         -1.54195562, -0.71400818],
       [-3.60996401, -3.38076397, -1.5714142 , ..., -0.9724252 ,
         -1.37485687, -0.6149685 ],
       [-4.69144366, -2.82302625, -0.75307605, ..., -0.18453999,
         -2.36241681, -0.2712815 ]])
```

We can see that we got the same transformed data using either SVD or PCA.

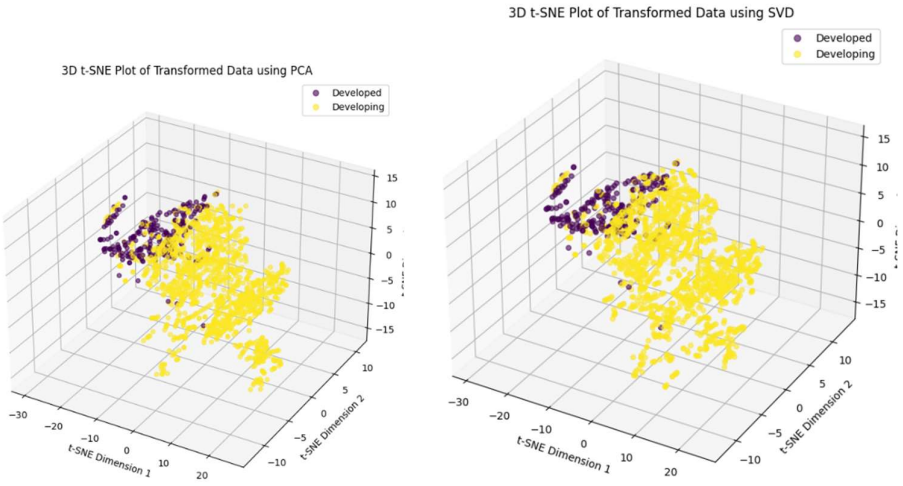
C. Visualize the data (t-sne plot) after applying SVD and PCA

A t-SNE model is created with two components, indicating a two-dimensional visualization.
Fit and Transform Data Using t-SNE:
The t-SNE model is fitted and used to transform the data into a lower-dimensional space.
Map 'status' Labels to Numeric Values:
The original categorical labels ('Developed', 'Developing') are mapped to numeric values (0, 1) for color-coding.
Create Scatter Plot:
A scatter plot is generated with points in the t-SNE-transformed space. Points are colored based on the numeric labels and have reduced transparency.
Plot Configuration:
The plot is configured with a title, and axis labels for the two dimensions.
Add Legend:
A legend is added to the plot, indicating the correspondence between numeric labels and the original categorical labels.
Show Plot:
The final t-SNE plot is displayed for visualization.



Similarly a 3D tsne scatter plot is plotted.

Here we use top 3 principal componenets to plot the graph using developed and developing sample data points.



D. State your conclusions as to how SVD and PCA have helped here

- SVD and PCA are effective techniques for reducing the dimensionality of the dataset while preserving essential information. This is particularly useful in our dataset with a large number of features in the dataset.
- The cumulative explained variance plot obtained through PCA assists in determining the number of principal components needed to retain a certain percentage of the total variance. This aids in balancing dimensionality reduction with the retention of information.
- The transformed data obtained after applying SVD and PCA can be used for further analysis and interpretation. We can build better performing model using the transformed data
- Initially when we plot pair plot between features we could see that feature vs feature plot was very difficult to fit a binary classification model. But the TSNE plot using top two principal components has a clean boundary and its easy to build a classifier model on this transformed data using principal components.
- These principal components can be used as features to build better and efficient models.

```
using original data
Accuracy: 0.7998
Confusion Matrix:
 [[ 208   31]
 [ 296 1098]]
Classification Report:
              precision    recall  f1-score   support

   Developed       0.41      0.87      0.56       239
  Developing       0.97      0.79      0.87      1394

    accuracy                           0.80      1633
   macro avg       0.69      0.83      0.72      1633
weighted avg       0.89      0.80      0.82      1633
```

```
using transformed data
Accuracy: 0.9273
Confusion Matrix:
 [[ 73  25]
 [ 23 539]]
Classification Report:
              precision    recall  f1-score   support

   Developed       0.76      0.74      0.75        98
  Developing       0.96      0.96      0.96       562

    accuracy                           0.93       660
   macro avg       0.86      0.85      0.85       660
weighted avg       0.93      0.93      0.93       660
```

- I have built an SVM classifier to test the accuracy and we see that our model performs better when trained with transformed data with reduced dimensions.
- In summary, the application of SVD and PCA to the WHO dataset has provided a means of dimensionality reduction, visualization, and insights into the relationships between features and life expectancy categories.