Preprocessing

Before any classification algorithms were performed, two data frames were created of the **world.csv** and **life.csv** files. The data frames were merged based on country code, removing the countries in the world dataset that did not contain a life expectancy as these countries would provide no use to build a model out of. A train/test split was performed on the data using the **train_test_split** function from the "sklearn" library, with 2/3 of the data allocated to the training set ,1/3 for testing and the random state parameter set at 100. The life expectancy was used for the target variable. It was noticed that there were missing data entries in the world dataset so median imputation was performed using the median of the training set. This was performed on all the features on both the training and testing set to ensure that no data from the test set "leaks" onto the training set which is used to form the model hence preventing an overfitted model. Both the training and testing features were then normalized using sklearns **StandardScaler** function and was fitted according to the training data. This scaled the data to unit variance and removed the mean so the classification algorithms would perform more accurately because all the features would be weighted evenly.

Task2A

Evaluation

With k=5, k-NN gave an accuracy score of 0.820, k=10 gave a score of 0.869 and decision tree only produced a final score of 0.705. Since the decision tree classification was performed with a random seed, the accuracy score oscillated in the 0.67-0.75 range every time it was performed and did not take on a stationary value making it not very reliable. k-NN with k=10 performed better than k=5 which is not surprising given that a higher value for k would make the classification less suggestive to noise points. The dataset is fairly large, and a higher k value would mean a more accurate representation of the data would contribute to the classification. A basic rule for choosing a k value is; $k = \sqrt{n}$, where n is the training sample data and since there were 122 training samples, this method for choosing k better approximates 10 not 5.

Task2B

In this task PCA and feature engineering was used to improve the performance of 5-NN classification in order to better predict life expectancy of countries. Three different implementations of 5-NN were tested, each using a different method of manipulation to the original set of 20 features in which all resulted in reducing the dimensionality of the data to 4 final features to perform 5-NN on.

First Four Features

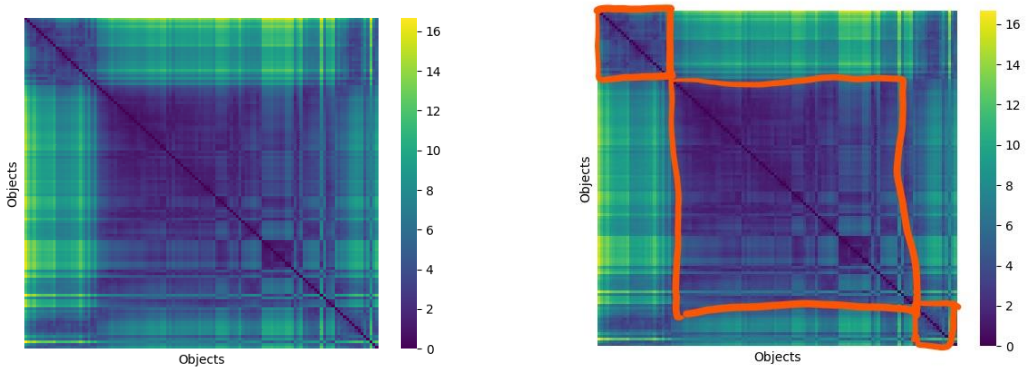
Since the dataset used for the k-NN in task 2a is the same the data used here, the merged data frame was manipulated to only pass in the first four features into the **train_test_split** function with the training data consisting of 2/3 of the full dataset. Other than this, the process for performing the classification is very similar to task 2a. Imputation on the training and testing data was performed using the training sets median and the features mean was removed and scaled to unit variance. 5-NN was then performed on the testing and training data consisting of the first four features listed in the **world.csv** file.

Principal Component Analysis (PCA)

After the data was split into 2/3 training and 1/3 testing, median imputation was performed on the training and testing data. The data was then normalized using the same functions and method as the first four feature 5-NN classification did. It is necessary to scale the data before performing PCA because since the projections are based on the variance of the features, ones with higher variance will have a higher weight in the analysis and thus normalizing is necessary. PCA was performed using the **PCA** function by importing **decomposition** from the sklearn library with the number of components equal to 4. The analysis was fit with the training data and transformed both the training and testing set. 5-NN was then performed on the four features chosen using PCA.

Feature engineering

Just as the other methods, the data was split, imputed and scaled the same way. The VAT function was used with the training data to visually assess how many clusters were to be used for the k-means clustering which generated a new feature “Cluster Label”. Scaling occurred before the visualization in order to control the effect of variance in the dataset. The figures shown below shows the visualization and 3 clusters were visually assessed from the output, which is in accordance with the discrete target, Life Expectancy (Low, Medium, High). Thus, k-means clustering was performed to generate 3 clusters each with their own centroid. Each point in the testing set was assigned to its nearest cluster using its closest centroid in order to create the “Cluster Label” feature. The interaction term pairs were generated from the unscaled 20 original features, producing 190 features. A total of 211 features were produced but the **SelectKbest** function was used from the sklearn library to reduce dimensionality and the top 4 features that produced the highest mutual information gain were chosen for 5-NN classification.



Evaluation:

Out of the 3 methods, feature engineering performed the best with an accuracy score of 0.836, second best was the first four feature method with a score of 0.770, and PCA performed the worst at 0.754. Its unsurprising that the feature engineering method performed the best given that there were 211 features to choose from based on their mutual information to the life expectancy, which works well for non-linear data. However, PCA does not work as well with non-linear data and thus a probable explanation as to why it was not as well performed. It is surprising though that the naïve method of choosing the first four features performed better than PCA did. To better improve the performance and accuracy score of 5-NN, more detailed feature engineering and interaction of features such as the ratio between certain features could be applied to find features with high mutual information. This would create greater variety in the final features used for 5-NN classification. Another suggestion would be possibly to implement dimensionality reduction that accounts for non-linear data as a replacement of PCA and hence possibly improving the classification method. It can be said that the classification algorithms are reliable given that the imputations made sure no data leakage occurred and there was not a significant portion of the features missing. The most reliable methods would prove to be 10-NN method in task 2a along with the feature engineering method in task 2b and they produced the highest accuracy scores. The k-NN method with k=10 was a good choice for the k value in the algorithm and the careful feature selection for the feature engineering method allowed for a relatively high scoring classification performance