**Task 2a**  Data was first pre-processed through median imputation of missing values. Each dataset was sorted separately based on country code due to ambiguity of country names and merged. The sorting allows for easier merging and subsequent classification.

Two supervised algorithms, Knn, using k=3 and k=7 and decision tree, were used to predict the class label (life expectancy) of each country. A max depth of 2 was chosen for decision tree to avoid overfitting and maintain node purity. For the K-NN model, data was standardized prior to classification (mean removal and divided by unit variance).

Of the three methods, the K-NN (k=3) performed the best (A=0.750), followed by the decision tree (A=0.708) and the kNN results with k = 7 performed worst (A=0.646). Knn (K=3) performed better than the decision tree because it uses the Euclidean distance function to compute distances between data points that is suitable for homogenous continuous features in this data. The slightly poorer classification performance of decision trees may be because even with the use of pre-pruning of trees (max\_depth = 2), the tree still tends to overfit and does not generalize well. The max\_depth might still need analysis. From statistics computed, differences between feature variances for are really large (max 375929954.006, min 1.79). Since decision trees did not require data normalization, it might be biased toward features with more variance. In contrast, due to the data standardization in K-NN, it does not face this issue. K=7 model performed worse than K=3 because the greater k value reduces the inter-cluster variance distance and smoothens the cluster boundaries. The neighbourhood may include points from other clusters, resulting in increased bias.

**Task 2B Method** Data of 20 original features was first split into 70% for training and 30% for testing. Featuring engineering and selection was conducted using 2 methods.

In method 1, 190 interaction term pair features were created for training and testing sets separately using *PolynomialFeatures*. One additional feature, the cluster label, was engineered using K-means-clustering (k = 3). The Elbow method was used to determine the value of K. The relationship between the number of clusters and Within Cluster Sum of Squares (WCSS) was graphed, and the value of n where the change in WCSS begins to level off (elbow point) is chosen as K value(task2bgraph1.png). This method examines the percentage of variance, the ratio of the intra-cluster variance to the total variance. K=3 was chosen because adding another cluster doesn’t decrease the within-cluster variances and thus create little information gain to the clusters.

Next, the data was preprocessed using standard scaling(mean removal and divided by unit variance). The cluster label of the training and testing data (0, 1 or 2) was separately predicted and assigned using K=3 and KMeans.

Random forests were used for feature selection. This algorithm works by first, randomly drawing samples from the original data; second, for each group of samples, grow a classification decision tree, randomly sample features and choose the best split from among those features based on their gini impurity. Features that cause the least increase in impurity at each node were removed. Each tree ranks features by placing nodes with the least increase in impurity at the root of the trees, notes with the greatest increase in impurity at the root of trees. Thus, by pruning trees below a particular node using the *SelectFromModel* function, we can select the four features that increase the purity of the leaves the most, i.e. the four most important features. Lastly, the four most important features were selected by aggregating the prediction results of the numerous decision trees that confers reliability.

Random Forests were used for this dataset because the data was all numerical. With randomizing mechanisms in both choosing records and feature selection, this method guarantees that the trees are de-correlated and therefore less prone to overfitting. This can be seen in the pair plots (task2bgraph6.png), only minor feature correlations were captured (between F3 and F1). The selected four features have the highest importance, as validated through comparison of two bar charts (task2bgraph2.png and task2bgraph3.png).

In method 2, Principal Component Analysis (PCA) was used to transform the input data containing 20 features by projecting it into 4 principle components. The training data containing the 4 PCs was standardized (mean removal and divided by unit variance) and subjected to K nearest neighbour (k=3) model for prediction. The K-nn classification accuracy scores for both methods were compared to results of a training dataset with sampling the first four features of the original features.

**Task 2B Results**

Feature engineering using interaction pairs and feature selection using random forests increased the classification accuracy (A=0.792). This is because generating interaction features can detect the interdependence between features, which is more accurate than the original features that are assumed to be independent and thus prone to noise. Also, the random forest algorithm effectively selects the 4 most important features by averaging the results of numerous decision trees. It greatly reduces feature correlations and eliminates useless, noisy features, leading to high accuracy and reliability.

PCA also increased prediction accuracy (A = 0.729). By using orthogonal transformation to convert data to 4 principal components, PCA eliminates irrelevant features, reduces noise learning and creates mutually independent new features. This can be validated by the high variance captured by the 4PCs, which captures 90.9% of total variance, as shown from the printed explained variance ([0.671, 0.174 0.044 0.020]).

Pair plots between the original four features, and four principal components were created. Task2bgraph4.png shows independence of 4 PCs from each other with weak feature correlations, as shown by the nearly flat scatter plots and distant data points. This contributes to the high accuracy of PCA since it predicts using highly independent, useful features and eliminates noise. This can also be shown in the clear decision boundary between the ‘low’ (blue) and ‘high’ (green) clusters and the non-smooth decision boundary between the medium cluster and other clusters in Task2bgraph4.png, which shows high model sensitivity.

In contrast, Task2bgraph5.png, pair plots between the original 4 features A-D, show high linear correlations between features A and D, and between features C and A. The decision boundaries between three clusters are also less clear than that of PCA. The highly correlated features confers the model a greater probability to miss out true explanatory features while adding noise. This justifies that PCA creates more important and independent features compared to original ones.

**Conclusion** Overall, the K-NN classification model is quite reliable since it predicts with quite high and consistent accuracy scores for each subset of selected features. Feature engineering and selection using both PCA and interaction pairs compounded with random forests yielded more accurate results than original features.

For future improvement, the median imputation of missing values in this experiment could have underestimated the true variance of the data since different columns have different rates of missing values. To improve the classification accuracy, we can use a model-based imputation, e.g. using KNN, to improve the data quality by introducing randomness, hence the resultant accuracy. Also, we can re-determine the number of features (rather than 4) used for classification to include only relevant and important features. Moreover, we can use k-fold cross validation to select the best train-and-test split and reduce bias in the training set by generating a cross validation matrix. The split ratio will be chosen such that it maximizes the classification accuracy. Tuning the parameters of KNN such as the value of K and the distance measure function could also improve classification accuracy. Lastly, we can use ensemble methods that use multiple learning algorithms and testing data will be classified based on the majority of the models' classification.