Task2a

Starting from pre-processing on data, since I am predicting the life expectancy, I first merge world and life datasets and exclude the countries without recorded life expectancy. I split resulted data table into train set and test set, where median, mean and std are calculated based on only the training set. Median imputation is done by replacing all missing values in training and testing set with median of training set for each feature, and so for scaling into unit variance with the function of StandardScaler which fit for training set. The mean, median and std for each column is computed in 'task2a.csv'.

After pre-processing, three algorithms firstly build the model based on pre-processed training data, and then calculate the accuracy by applying to the test data. As a result, the accuracy of decision tree ranks the last of 78.689%, then is knn classifier with k =5 of 80.328% accuracy and k=10 of 83.607% accuracy. Since all my input variables are continuous, knn works better than decision trees whose advantages emphasis on interpretation on categorical inputs. In comparison of number of k, larger values of k make smoother between the boundaries of classes yet present less sensitive to noise. Yet this fundamental flaw is overcome by normalization in pre-processing. It eliminates the noises and so let k-10 performs the best.

Task2b

For comparing different feature select methods, the data is imputed with median and normalised as exactly the same in task2a. In this case, normalization helps to eliminate the noises in features which cause incorrect classifications and improves the overall accuracy

The first selection method is using interaction term pairs and clustering labels. The number of clusters I select is 16, which is printed as standard output in my program. This is decided as it gives the highest accuracy when comparing all possible accuracies implementing k clusters (k ranges from 2 to 102). A graph is also produced and named "task2bgraph1.png" for visualisation. After generated features are add, there are 211 features in total. The data is split into training set and testing set, where each feature is firstly cut into bins (discretisation), then compute the mutual information. It is calculated based on entropy, a measure of impurity/homogeneousness of records in split group. In this case, I aim to choose four features with highest MI, so that they have relatively homogeneous distribution.

Another selection method is called PCA, which takes the first four principal components. It is especially useful when the dimensions of the input features are high (e.g. a lot of variables) As for this method, the importance of each feature is reflected by the magnitude of the corresponding values. The larger they are these absolute values, the more a specific feature contributes to that principal component. In sklearn, the components are sorted by explained variance, which is maximised after PCA

Last but not the least, is the method first four features. It just takes the first four features from the original dataset. The result accuracy is inconsistent due to uncertain display order of features. If the features are accidentally important, its accuracy would be significantly high, at least higher than the baseline (accuracy of directly perform knn5 to all features). Yet if four least important features are displaced as front columns, the resulted accuracy would be even lower than the baseline.

In this case, the accuracy of selecting first four features is 73.770%, even lower than the baseline of 81.967% in terms of using same knn5 classifier. In comparison of accuracies using PCA and feature engineering, they are about the same. PCA is sensitive to scale and usually gives more weight to data with higher order of magnitude. It reduces dimensionality by exploring how one feature of the data is expressed in terms of the other features (linear dependency), thus is beneficial when the dimensions of the input features are high (e.g. a lot of variables) in this case. Feature engineering instead, takes the target into consideration. It will rank your input variables in terms of how useful they are to predict the target value. This is true for univariate feature selection. The result accuracies here are not so reliable as the best k for cluster label is not consistent and the accuracies changes all the time. It will be more reliable if there are fewer missing values and importance of features varies greatly. To improve that, I can also implement additional algorithm such as EM algorithm to build complex statistic model and find maximum-likelihood estimates for model parameters.