Task2c

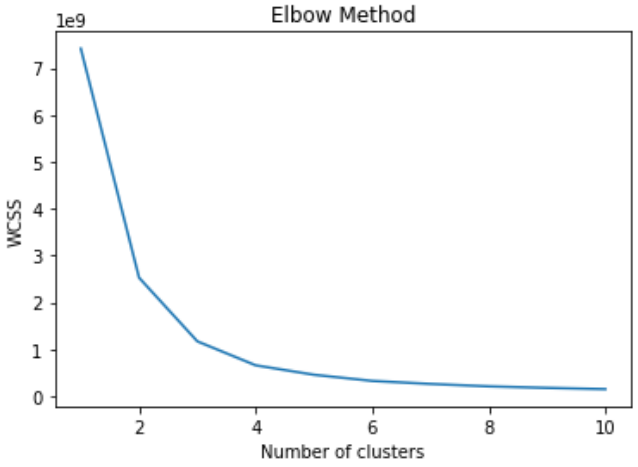
Task2a

k-nn with k=10 is performed better compare to the other two algorithms. I first merge the dataset of world and life together, choose only the features and class label, remove those countries that do not have a label for life expectancy as it is not helpful to train our model. Let the life expectancy in the remaining 183 datasets to be the class label and split 2/3 of the data to be the training set and the remaining to be the test set. Replace each missing value in both training and testing set to be the median of its corresponding feature column accordingly. Set all values in each feature to be float. Then I normalise the data to have 0 mean and unit variance for later comparison. Followed by using method in sklearn lib to predict the class label of the test set use model fitted by the training set. Each class label for the test set is determined by the 5 nearest neighbours while k=10 is similar but this time the class label is predicted using 10 nearest neighbours which gives a higher precision of 0.869 compare to k=5 where the accuracy score is 0.820. However, larger k will not always result in a higher precision but rather we need to test to see which k is the best for the specific dataset. In addition, the decision algorithm will give a accuracy score of 0.787 which is probably not an optimal algorithm that we should could to model this training set. Furthermore, I get the median, mean and variance for each feature in the training data to a dataframe and save to a csv file.

Task2b

I first merge the file of world and life together, choose only the features and class label, then discard those rows without a class label. For each feature in the dataset, I replace the missing value with the median of the corresponding feature and set all values in each feature to be float. Then I get all the combination of features, get the multiplies of values in these features hence obtain 190 new features.

Furthermore, I use the elbow method by plot a line graph of number of clusters verses the sum of square distance within each cluster. Since we should choose a number of clusters that adding another cluster will not lower the wcss too much, from the graph *task2bgraph1* we can see that 3 would be the optimal number of clusters to choose. In addition, we have three distinct class label which is also reasonable for us to have three different clusters. I fit the dataset into a kmeans model in order to get the feature of *clusterlabel*.



After join all of the 211 features together, I split the 2/3 of the data into training set and the other 1/3 in to testing set. The method that I use to generate feature selection of the 211 features is to calculate the normalised mutual information for all of the features and choose the four features that has the highest NMI. I use this method as it measures how much information that each feature helps to correctly decide the class label, where higher NMI indicate that the feature will have more information gain. It is also helpful for the situation where our class label is categorical. After normalised the training and testing set, I fit the training set values to a 5-NN model and thus get the accuracy score of 0.770. The second method is also to split the 2/3 of the data as training set and 1/3 in as testing set. After normalising the xtrain and xtest, I then use PCA algorithm from the sklearn

library with number of component set to be 4 to do feature selection and obtain an accuracy score of 0.754. The third methods is the same in model training and calculating the accuracy score while the only difference is that it simply chooses the first four features from the dataset as feature selection and therefore get 0.754 as its accuracy score.

Feature engineering by selecting 4 out of 211 features including features produced by the interaction terms will get the best results. The 5-NN method have an accuracy score of 0.770 which is better in this case compare to 0.754 for the PCA algorithm. Feature engineering of the 211 features allows us to select from more features which might be better for the classification. Furthermore, since the third method is simply selecting the first four features without any algorithm applies, since the features are randomly selected, it will produce a precision (0.754) that is less or equal to the other two methods where different algorithms has been applied to choose the important features.

In conclusion, to some extant my model is reliable. I get 0.77, 0.754 and 0.754 for the 5-NN classification using three different feature selection methods respectively which are acceptable accuracies but still have improvement. A technique to improve the classification accuracy could be increasing the size of the training set. More data is used to train the model, more precise the classification could be. We could also use k-fold cross validation to repeat the modelling 10-100 times in order to reduce any bias from random number choices. Furthermore, feature engineering is useful in a sense that it helps use to identify the important features while it also saves time in training the model. However, in another hand, it also results in some information loss as we have removed some less important features from our model training. This might be the reason why the accuracy scores that I obtained after the feature selection is lower than the score that I get when use all of 20 features from the dataset (82%). Nevertheless, it does not mean that feature selection is not applicable. While when there are hundreds or even more features in a dataset, it is almost impossible to do classification using all of the features, feature selection then becomes necessary in this case. Sometimes when we really need the most precise result possible and the number of features is quite small, it is considerable to use more features or even all of them for classification in order to have less information loss and hence obtain a more accurate result. Moreover, I only use 5-NN for the classification while there may be the best algorithm to choose. Different classification algorithm or different numbers of k for the knn method could make a large difference in the resulting accuracy. I should also try other methods such as decision tree or to alter the k for the k-nn method so I can compare which algorithm will give me the higher precision.