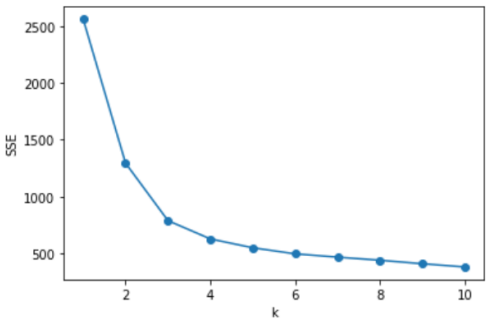
1. For Task-2A, k-nn(k=7) algorithm performed better on this dataset since it has the highest accuracy. For k-nn algorithm, k=7 performed better. Starting with the data pre-processing stage, I replaced all the '..' in csv files to 'np.nan' since it would be more convenient to do the missing data imputation. I merged and sorted two csv by the key of 'country code'. I split the dataset into target and class label. Then I applied 'train_test_split' to make 70% of the data set to be the training set while the rest to be the testing set. This is to reduce overfitting problem when evaluation performance of a classifier. To deal with the missing data, median imputation was applied to the training set and testing set. Since the test set is not known to us at any point during training. We only use it when evaluating a model. Therefore, I used the median of training set data to impute missing values in training set and testing set data. Then I scaled features by calling the function 'StandardScalar' which subtract its mean and divided by standard deviation of each column. The result data has mean of 0 and variance of 1 which avoid measurement deviation of distribution. This increases the accuracy of the model since knn is based on calculating the distance between data. Then I fit the training data to the decision tree with maximum depth of 3, knn(n=3) and knn(n=7) respectively. The accuracy is 0.709, 0.673 and 0.727 which tell us the percentage of right-predicted data in the testing set.
2. For Task-2B, it aims at extracting more useful data and features to improving the accuracy of the 3-NN model. I have applied the same data pre-processing steps as task1B, including replacing '..' to 'np.nan', splitting data in to training and testing sets, median imputation of missing values and standardizing data. Because k-NN algorithm is based on calculation distances between data points, standardizing is important. There are three different methods applied in this task. The first one is feature generation which creating interaction term pairs and adding clustering labels. After feature generaion, there are 211 features in total(printed). The method I chose to do feature selection is filtering by calculating mutual information. If the feature shares more information with the class label, it should be more useful for prediction. I calculated all the 211 features mutual information with the class label and selected 4 features with the highest mutual information(printed). The second method is PCA which reduce the dimension of the data. Since high dimensioned data become sparse and may lead to overfitting. I applied PCA to the training (printed)and testing set before doing classification. Last method is like randomly choose 4 features and compare previous accuracy with it.
3. For the method I used to choose number of clusters for the clustering label generation, I used the elbow method. The elbow method runs k-means clustering on the dataset for a range of values for k and I chose k to be in the range of 1 to 11. For each value of k, I computed the sum of square distances from each point to its assigned center which is stored into the SSE array(printed). Then I used line graph (task2graph1.png) to visualize the change of SSE according to the change of k. If there is a strong inflection point, it is a good indication that the underlying model fits best at that point since at this point SSE decreases a lot. As shown in 'task2graph1.png', k=3 is an inflection point, so I chose k=3.



4. I have applied one of the filtering method of feature selection which is the mutual information. I used 'mutual_info_classif' to compute standard mutual information between each feature with the class label and store the mutual information score into a dictionary with key of feature index. Then I sorted the dictionary and chose the features with the top four highest mutual information scores (as printed array). Then 3-NN algorithm was applied to the columns of the 4 features. The reason of this method works is that higher mutual information could tell us the corresponding feature shares more information with the class label which means that they are more related if the standard mutual information score is close to 1 and more independent if close to 0. Since more related feature to the class label would give more accurate result.
5. PCA has the best results for classification using 3-NN. Because there are only 128 countries with 20 features in the dataset and the dimension is high. Sparse data may lead to overfitting problems. Also, all the distances between pairs of points begin to look the same which impact K-NN accuracy since it is based on calculating distance. Therefore, PCA avoid curse of dimensionality.
6. Firstly, median imputation to missing values may not be a great choice here since some of the country may have high or low value of some features rather than median of the world like America of the dataset. Using median is not always proper for some countries. Therefore, I think it maybe better to fill the missing values based on data of similar countries. We could calculate k most similar countries using other non-missing features and predict the missing data based on these k countries. Also, choice of k is important, we could test the accuracy of model of different k and select the most accurate one.
7. The model is not too reliable since the dataset is small which could lead to bias. Because the sample size is small we could apply k-fold cross validation to test the model in order to reduce any bias from random sampling. Therefore, the selection of k could be done by doing k-fold cross validation since we can find the best parameter when evaluating different model. Also, 3-NN is not a great choice for this data since small k is sensitive to noise data.