Elements of Data Processing, 2020, Assignment 2 Task 2C: Classification Report
Name: Aoi Fujii

Task 2A:Comparing classification
Experiment method: Two datasets from world.csv and life.csv are merged based on the country code. The nan values is imputed to the missing value represented as '..' and the values are converted to the float type. Then, feature columns and its class label is extracted. 2/3 data is split to a training set and the rest to a test set. The median value of the training set is imputed to the training set and the test set. Both training set and test set are scaled by removing the mean and scaling to unit variance. The StandardScaler() from sklearn.preprocessing fitted by the training set is used for this process. Then each classifier (KNeighborsClassifier(k=5, 10) and DecisionTreeClassifier(with maximum depth of 4)) is trained using the training set. Lastly, the test set is applied to each classifier using .predict() and accuracy score is obtained by the accuracy_score().
Performance of each algorithm: Accuracy was 84% for 5-NN, 82% for 10-NN and around 69% for decision tree (the accuracy of decision tree ranges from around 66-77%). In this task, all the parameters other than maximum depth is set to defaults, but the random state could be set to a certain value to obtain a deterministic behaviour of the decision tree. K-NN performed better for this dataset compared to the decision tree. This could be because of the susceptibility of derision tree to outliers and loss of critical information while handing continuous variables. For K-NN, k=5 performed better compared to k=10. The figure 1 shows that the accuracy fluctuates around 0.84 for k=5 to k=25 and accuracy is around 2% higher for K=5 compared to k=10.

Task 2B: Feature Engineering and Selection
Experiment method:
Preprocessing: Firstly the data from world.csv and life.csv are inner joined by the country code and nan value is imputed to the missing value represented as '..'. Then the merged dataset is split to test set and training set by 1:2 ratio. For all the methods(feature engineering, PCA, first four), the median value is imputed for the missing value using SimpleImputer and data is scaled by the StandardScaler and then MinMaxScaler() on both the training set and the test set. Only the training set is used to fit the imputer and scaler so that the test result will not be biased. However, feature engineering for interaction terms pair is conducted before the scaling as standardisation will change the value of the generated feature.
Feature engineering:
Interaction term pairs: Features are generated by utilising the PolynomialFeatures from sklearn.preprocessing. The parameter, interaction_only, is set to True and include_bias is set to False to generate the data of the interaction term pairs and original features only. The training set is fit to this function and both test and training set is transformed to get all the 210 features (20 original and 190 generated). Then the values are standardised by StandardScaler and normalised by MinMaxScaler.
Cluster-label feature generation: The training set is fit to KMeans to generate a new feature obtained by the resulting cluster label. The feature for the test set is obtained by applying .predict() on kmeans. VAT is used to visualise the dataset to select the number of clusters. Figure 2 shows that there are 3 clusters for this dataset. This is further justified by the Elbow method shown in the figure 3. Sum of square distance between data points and cluster centroid starts to flatten after k=3 so 3 cluster is appropriate choice for this dataset. The generated feature values are shown in the standard output.
Feature selection method (from 211 features): Features from interaction term pairs including original features and the cluster-label feature are merged into one dataset. SelectKBest from Sklearn.feature_selection is used to get the best 4 features from 211. The test dataset if fit to SelectKBest and a list of index of selected features is obtained by .get_support(indices=True). Figure 4 shows the calculated score for each feature. The selected 4 features and their scores are shown in the standard output.

PCA: The training set is fit to PCA(n_components=4) and both training set and test set are transformed to get the 4 principal components from the data set. Then, KNeighborsClassifiers(n_neighbors=5) is trained by the training set and applied to the test set using .predict(). Accuracy is calculated using accuracy_score().

First 4 features: First 4 features from the dataset, which is shown in the standard output, is used for this method. The training set is applied on the KNeighborsClassifier(n_neighbors=5) and the test set is applied to find the predicted values from the classifier using .predict(). Accuracy score is calculated by the accuracy_score().

Conclusion
Reporting of the result:
The accuracy observed was 80.95% for feature engineering, 84.13% for PCA, and 74.60% for the first four features. The most probable reason for the highest accuracy of PCA is that PCA finds a set of features that gives the most variability of the data. Thus, most uncorrelated features will be selected which will result in the better representation of the dataset and higher accuracy score.
Interpretation, justification of the result and reliability of the method:
PCA: The variation for each principal component is 0.79, 0.06, 0.03, 0.02 as shown in the output. This means around 90% of the information is retained by these 4 components and 10% is lost while reducing the dimension from 20 features to 4. In addition, the predicted model of PCA in figure 5 shows that there are the less correlation between features. Thus, the

selected features represents most of the information from the original dataset. Furthermore, the figure shows the clear separation of 3 classes based on the life expectancy. Overall, PCA worked better to represent this dataset.

Feature generation: This method recorded 80% which is higher than choosing the first 4 but lower than PCA. The features with highest 4 scores are selected by SelectKBest based on the selected scoring function. However, since one of the 210 feature is categorical while others are continuous, this could lower the accuracy score as a measure. In addition, there are limited choice of scoring function for SelectKBest. More variation of different scoring function could improve the scoring measure to make it more suitable for this dataset.

First 4 features: Figure 6 shows that most of the selected 4 features have correlations between each other. For example, cause of death and birth rate have a strong positive correlation. Since the features are selected at random, they are not necessarily a good measure to predict the test set (i.e. having a high correlation between features). Therefore this method is less reliable and recorded the lowest accuracy of the three method.

Suggestion for improvement:

First suggestion to improve classification accuracy is to address the missing values in a different way to improve more accuracy. For this method, median value was imputed for the missing value but the median is not always close to the actual value of the dataset. The missing value could be predicted from the other data with similar trend in their feature value. Moreover, predictions of multiple classifiers could be combined to give more accurate result. To ensemble the classifiers, they should be trained on the same dataset and the one with the highest confidence could be used for prediction. Lastly, cross validation could also be effective as it avoids biased sampling. This gives more accurate measure of the classification to improve the methodology.

Figure 1. varying k versus the prediction accuracy for K-NN

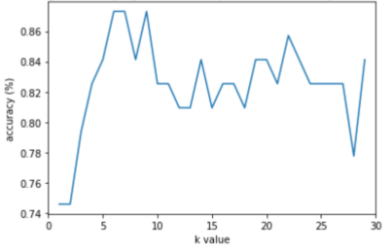


Figure 2. VAT for the selected 4 features after feature engineering

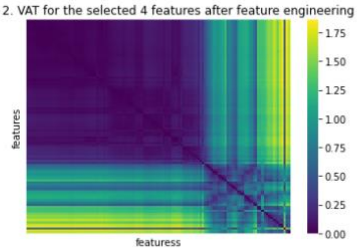


Figure 3. Kmeans for differnt number of clusters

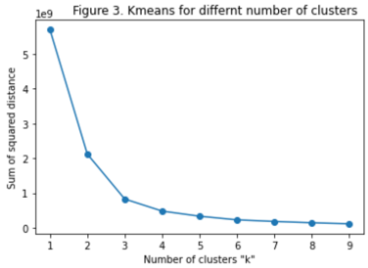


Figure 4. feature score for feature selection

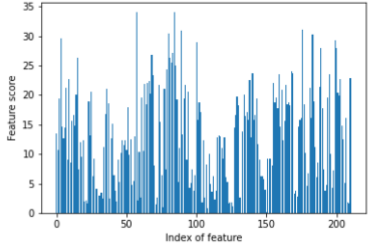


Figure 5. Principal component analysis of life expectancy at birth

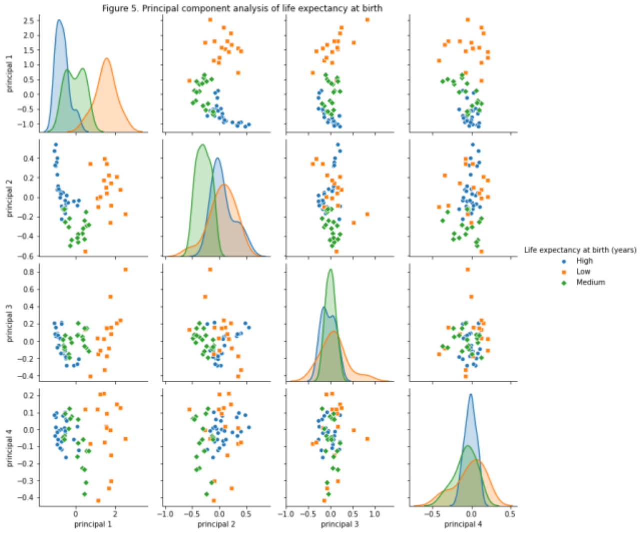
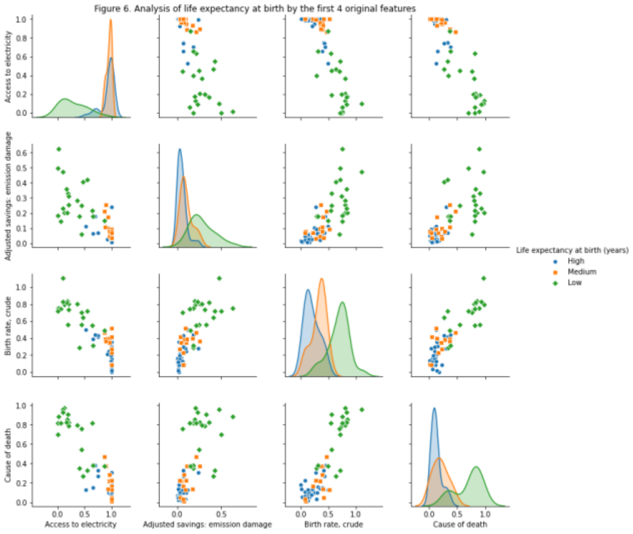


Figure 6. Analysis of life expectancy at birth by the first 4 original features



Reference:
COMP20008 workshop week8: https://canvas.lms.unimelb.edu.au/courses/12012/files/3342631/download?download_frd=1
COMP20008 workshop week9: https://canvas.lms.unimelb.edu.au/courses/12012/files/3418960/download?download_frd=1