Task 2C – Classification Report

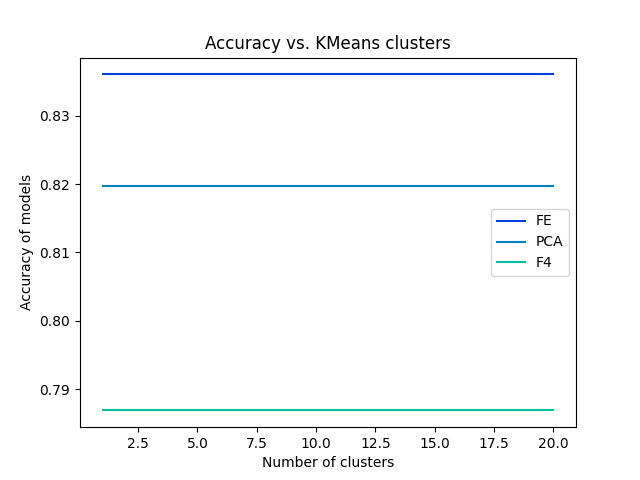
Task 2A

In task 2A, the decision tree algorithm acquired an accuracy of 75% while 5-NN and 10-NN achieved 87% and 80% respectively. Overall, the kNN algorithm performed better on this dataset, with the 5-NN being best of the two. The experiment was straightforward, starting with specifications on splitting the data (1/3rd testing data), imputing the train and test set with median values(used Imputer), and scaling each feature by removing the mean and scaling it to unit variance (using Standard Scaler). Then we used scaled data to make predictions with our respective models. With this, we also stored our mean, median, and variance for the dataset for which we imputed and scaled – for every feature.

The data cleansing and splitting process was standardized to compare apples to apples amongst the two models. And surely, kNN was better than decision trees since perform poorly with high data dimensionality.

Task 2B

Task 2B focused on feature engineering and selection with kNN as the standard model for all predictions. The feature engineering was split into 3 parts – interaction pairs and clustering, PCA (Principle Component Analysis), and no feature engineering.

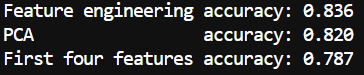
As seen in the output of the code, 210 features were created with the help of the Polynomial features function and 1 feature from kMeans clustering. The polynomial feature function returned 210 features, with features such as f1xf1, f2xf2, ..., fnxfn untouched and other terms such as interaction pairs of f1xf2 and so on. Similarly, the 211th feature is seen by clustering, which was fit from the processed world dataset (imputed and merged with life dataset). This was simply done by passing it through kMeans to fit in the model. Although, kMeans clustering had an important argument, i.e. the number of clusters. To choose the ideal number of clusters, I ran a for loop from 1 to 20 clusters to find the maximum accuracy and the best fit for the number of clusters. However, the code does not use kMeans clusters as a reliable feature (as seen below) – this was decided using the Random Forest Classifier. 211 features were passed through the classifier out of which a maximum of 4 features were selected using the Select from Model function. This gave us the top 4 features from the 211 to choose from. The random forest classifier achieved the highest accuracy of the lot. Seen as FE (Feature Engineered in the graph).

Note: Only FE is given the 211 features, other accuracies are depicted for comparison purposes only.

And therefore, as a default value we chose 1 to be the number of clusters since kMeans has a time complexity of n2 which would make the rest of the code a little more efficient.

The Select from Model function helps select features based on the importance of weights placed by the Random Forest Classifier. This model was used since we were to select 4 best features in a principled manner and one way to do so, like PCA, was to allow the model to pick out the most important features in its own unique way rather than a random pick of features or an educated guess.

The three methods provided the following output –



The Random Forest Classifier proved to be the best, followed by PCA, and then the first four features. This might be due to the fact that the Random Forest Classifier is a bagged ensemble model and with 211 features, it had a plethora of options to bag and choose to provide the best outcome as compared to a meagre 20 features given to PCA. And the first four features held less value as it was not a strategic pick from the lot.

Some other techniques that would help improve the data is better cross validation, bootstrapping and grid search CV. Cross validation is a resampling method used to evaluate machine learning models. With the number of columns lower than the number of features, it would be helpful to use such sampling methods to get more accurate results. Bootstrapping is another resampling method that uses random sampling with replacement to sample estimates. These resampling methods are important given the domain of the problem. There are only 183 countries with relevant data, and you would have to wait another year for more data. Rather if we resample and rearrange our data, we could achieve better results. And finally grid search allows use to use the best parameters across all models, since there were various models that had its own parameters out of which a unique combination could be formed for the best result.

This classification model is quite reliable given that the domain has large features with low number of rows, and the feature engineering mimicked exactly that kind of data. PCA allowed for dimension reductionality, and the Random Forest Classifier bagged several features that worked best for the data to both come up with 80+% accuracies. The data similarly is highly reliable since it comes from the world bank, and the world health organization proving that the model has been trained accurately.