Milestone2 Project Report

Part A Supervised Learning

1. **Generative approach**

1.1 Linear Discriminant Analysis (LDA)

1.1.1 Workflow of the source code

1) **Loading / Cleaning the Dataset**
   The CSV files of both train and test data are loaded using Pandas library. The WikiLarge_train dataset is cleaned by removing non-ascii characters, "-lrb-" and "-rrb-", and unnecessary space within each text. It is then split into the train and test dataset, 80% and 20% respectively.

2) **Creating BOW / Tf-Idf Vectors**
   CountVectorizer and TfidfVectorizer from sklearn are utilized for feature vectorization. The vectorizer is trained with the train dataset and used to transform the test dataset.

3) **Tuning Hyper Parameters**
   Hyper parameters tuning is done manually as well as using machine learning pipelines with GridSearchCV. In the pipeline, there are 2 main steps: preprocessing and model prediction. Within the preprocessing step, customized transformers such as ColumnSelector, DataTransformer are included together with the standardized text vectorizers (BoW and Tf-Idf) and feature selector (SelectKBest). Max_feature, ngram_range, min_df and max_df in the text vectorizer and k in the feature selector are the important parameters to tune.

4) **Training and Evaluation**
   After finding the optimal parameters mentioned above, the LDA model is trained using the train dataset. The classification report that summarizes accuracy, precision, recall, f1_score and support is generated using the test dataset in order to evaluate the classification performance. ROC curve is also plotted to illustrate the diagnostic ability of the binary classifier.

1.1.2 Learning Methods

Since text-based data is usually high-dimensional and sparse, it is necessary to use dimensionality reduction techniques in the pre-processing and model training. Linear Discriminant Analysis (LDA) achieves this goal by learning summary statistics from the input data and making predictions by estimating the probability that a new instance belongs to each class. Different from PCA, LDA is a supervised learning method and computes the directions ("linear discriminants") that will represent the axes that maximize the separation between multiple classes.

1.1.3 Feature Representations

Bag of Words (BoW) represents text as input feature vectors by describing the occurrence of words. TF-IDF is one step further than Bow by measuring information gain. These feature representations were chosen because they are commonly used and easy to interpret.

1.1.4 Tuning parameters

The objective of the initial manual tuning is to establish the plausible range of values for each parameter so that the pipelines can be trained efficiently without running out of memory space and taking too much time. Because the training dataset has close to half a million instances, text vectorization generates a sparse matrix of over 130,000 dimensions without any parameters tuning. Given that LDA can only accept a dense matrix as input, it is impossible to train a dense matrix of the same dimensions due to limited memory. Tuning parameters in vectorizers is the first step in eliminating infrequent features, followed by feature selection methods such as SelectKBest in which parameter k represents the number of columns to retain.

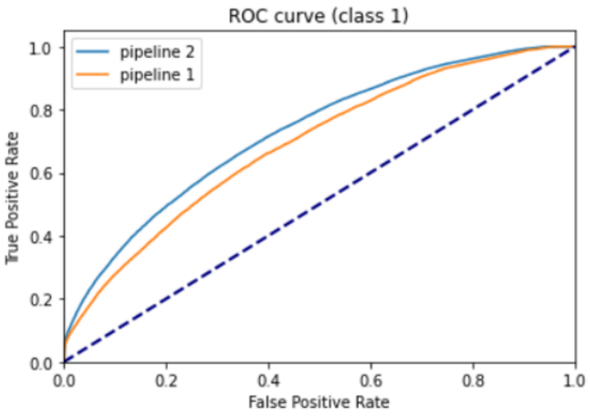
1.1.5 Challenges

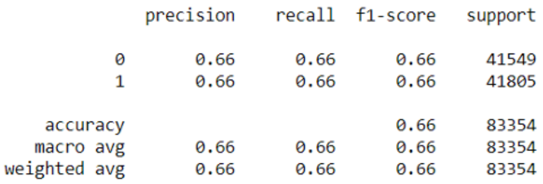
There are two challenges facing parameter tuning. First, five parameters with a minimum of two different values each lead to 32 models to tune each time. It takes at least 40mins depending on the value of max_feature. The higher the max_feature, the more information extracted from the data, allowing for better accuracy of classification. However, on the downside it requires lots of computational power and time to train the models. The second challenge involves building a pipeline that allows customized preprocess applied to the vectorizers. For instance, adding lemmatization or stemming to TfidfVectorizer is allowed by passing a function to the parameter preprocessor. However, it fails when such a vectorizer is added to the pipeline due to “serialization error”. In other words, the pipeline can no longer be run in parallel with GridSearchCV, which is a significant obstacle given a large dataset.

1.1.6 Evaluations

1.1.6.1 Evaluation Metrics

The highest accuracy score after hyper parameters tuning using pipelines is about 66%. Since our training dataset is balanced, accuracy can be considered as an appropriate metric to measure the performance of classification. Pipeline 1 and pipeline 2 are trained referencing the result from the manual tuning. Based on the ROC curve below, pipeline 2 gives the better performance as the curve is above the diagonal line and pipeline 1.



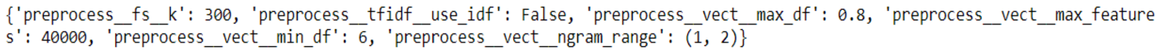


```
              precision    recall  f1-score   support

           0       0.66      0.66      0.66     41549
           1       0.66      0.66      0.66     41805

    accuracy                           0.66     83354
   macro avg       0.66      0.66      0.66     83354
weighted avg       0.66      0.66      0.66     83354
```

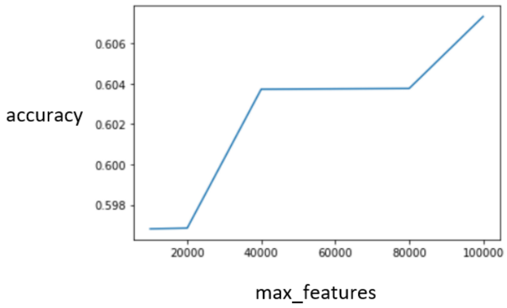
Below are the optimal parameters found by GridSearchCV

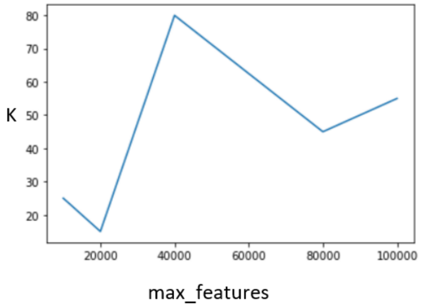


```
{'preprocess__fs__k': 300, 'preprocess__tfidf__use_idf': False, 'preprocess__vect__max_df': 0.8, 'preprocess__vect__max_feature
s': 40000, 'preprocess__vect__min_df': 6, 'preprocess__vect__ngram_range': (1, 2)}
```

1.1.6.2 Feature sensitivity

From the result of manual tuning, it is evident that as the number of max_feature increases, accuracy rate increases, but max_feature and k value do not have the same positive correlation: as max_features increase, the optimal k for feature selection varies. Below graph demonstrates these relationships.





1.1.7 Failure Analysis

Two failures occurred using the LDA model. First, “serialization error” using customized lemmatization and stemming functions in the pipeline is not solved. The manual tuning shows that adding these functions actually decreases the accuracy and increases recall. These additional text filtering functions are not used in the pipeline. Second, the accuracy score can not be further improved beyond 66%. Perhaps the dimensionality reduction and feature selection process might have failed to capture the most useful information in separating two different classes.

2. Discriminative approach

2.1 Bidirectional LSTM

2.1.1 Workflow of the source code

1) Loading / Cleaning the Dataset

In this stage, the CSV files(WikiLarge) are loaded using Pandas library. The loaded datasets are then cleaned based on the metrics below:

a. Drop irrelevant texts such as ‘-lrb-’, ‘-rrb-’.
b. Encode / Decode with ascii and drop errors.
c. Remove space tags / space at the beginning and the end of each sentence.

The cleaned WikiLarge_train dataset is then split into the train, validation, and test dataset. Tokenizer from Keras library is utilized to create pad sequences, trained with the split train dataset. These pad sequences are later fed to the deep learning model.

2) Creating Pre-Trained GLoVe Embedding Vectors

The pre-trained GLoVe is downloaded and unzipped. glove.6B.200d.txt, which contains 200-dimension embedding vectors for each word, is chosen and loaded as a hash-map. After that, embedding vectors of pre-trained GLoVe are mapped to the vocabularies of the training dataset. The mapped embedding matrix is later used to build the embedding layer in the biLSTM model structure.

3) Structuring the Model

In this stage, the model is structured using Keras Layers object. The model starts with the input layer, followed by an embedding layer, two Bidirectional LSTM layers, and an output layer.

4) Training and Evaluation

The model is compiled with ‘adam’ optimizer, ‘binary cross entropy’ loss function, and ‘accuracy’ metric. After compilation, it is trained with the train dataset and uses the validation dataset to monitor validation loss and validation accuracy. The process takes 5 epochs, but the Early Stopping method based on the validation loss is utilized to prevent overfitting. After the model is trained, it is then evaluated using the test dataset. Confusion matrix, accuracy score, precision score, recall score, f1 score, AUC score, precision-recall curve, and roc curve are used to evaluate the model. The training curve is used to see if there’s overfitting.

2.1.2 Learning Method

Bidirectional LSTM(biLSTM) layers are used as a learning method. BiLSTM is a way of utilizing LSTM, which overcame the limitation of RNN using gates, not only with the forward-directional way but also with the backward-directional way. The biLSTM is chosen because it is known for working well with the sequential data by using gates to prevent vanishing gradient problems.

2.1.3 Feature Representation

Pre-trained GLoVe is used as a feature representation. The GLoVe is a type of word embedding based on the words' probability of co-occurrence. The information on the ratios between each word is considered at the corpus level by utilizing the window co-occurrence matrix. The GLoVe embedding is used as a feature representation because it is known as a good way to represent the semantic relationship between words and thus would be a good combination with the biLSTM.

2.1.4 Tuning parameters

In order to adjust the complexity of the model, the number of biLSTM layers is tuned. The goal here was to see if there's any trade-off between the complexity and the effectiveness(training time and accuracy were taken into consideration) of the model. This was done by simply adding and dropping layers in the model.

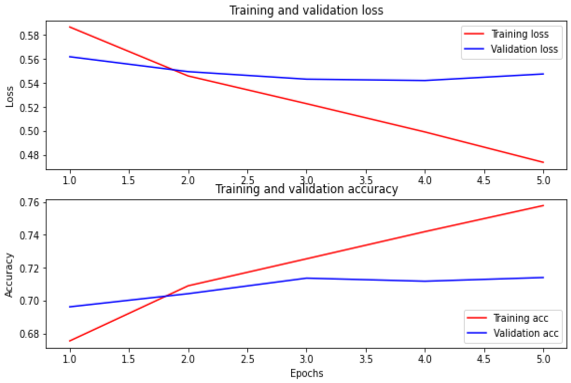
2.1.5 Challenges

It took a tremendous amount of time to train biLSTM layers, which often resulted in losing connection with the runtime. In order to solve the problem, the GPU provided by Google Colab was used to boost the speed.

2.1.6 Evaluation

2.1.6.1 Training Curve

Looking at the training curve of the model, the third epoch returned the optimal training-validation accuracy. However, no significant differences in validation loss and accuracy were found from each epoch.


Training and validation loss

Training and validation accuracy

2.1.6.2 Evaluation Metrics

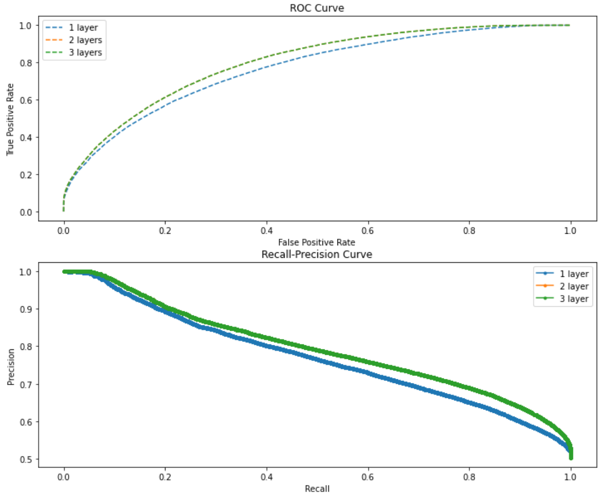
The model returned the test accuracy of 71.5%, which is slightly better than the baseline score provided by the faculty. Precision, Recall, F1, and AUC scores are also gathered.

Index	Score
Accuracy	0.7153
Precision	0.7016
Recall	0.7512
F1	0.7256
AUC	0.796

2.1.6.3 Hyper Parameter Sensitivity

As mentioned earlier, the complexity of the model was adjusted by adding and dropping the number of biLSTM layers.

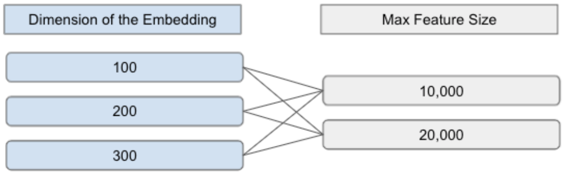
Index	1 layer	2 layers	3 layers
Training Time	31ms / step	64ms / step	98ms / step
Accuracy	0.6927	0.7153	0.7192
Precision	0.6902	0.7016	0.7181
Recall	0.7015	0.7512	0.7236
F1	0.6958	0.7256	0.7209
AUC	0.767	0.796	0.800



2.1.6.4 Feature Representation

1) Implementing Different Feature Variables

Two variables, the dimension of the embedding vectors and the max feature size, were considered when applying feature representation.



2) Sensitivity Test

It turns out that there wasn't any significant difference in the accuracy of different feature representations. However, some of them returned significantly better results than others regarding precision and recall scores.

Accuracy		Dimension of the Embedding		
		100	200	300
Max Feature Size	10,000	0.71	0.71	0.72
	20,000	0.71	0.72	0.72

Precision		Dimension of the Embedding		
		100	200	300
Max Feature Size	10,000	0.67	0.7	0.7
	20,000	0.72	0.7	0.71

Recall		Dimension of the Embedding		
		100	200	300
Max Feature Size	10,000	0.81	0.75	0.76
	20,000	0.71	0.75	0.76

F1		Dimension of the Embedding		
		100	200	300
Max Feature Size	10,000	0.73	0.72	0.73
	20,000	0.71	0.73	0.73

3) Identifying the Important Feature

As mentioned above, adjusting features didn’t return any significant difference when it comes to the accuracy and F1 score. However, what’s interesting is that when switching the dimension of the embedding to 100, the variance of precision and recall between both 10,000 and 20,000 max feature sizes were significantly greater than those of other dimensions.

4) Identifying the Important Trade-offs

When applying a different number of layers in the model, the training time went up almost linearly every time the layer was added. Other metrics, such as accuracy, weren't that simple. All of the scores increased when the model switched from 1 layer to 2 layers, while some of them increased and others decreased when the model switched from 2 layers to 3 layers.

2.1.6 Failure Analysis

When applying tf-idf created from sklearn library as a feature representation instead of GLoVe, the deep learning models kept using all the available memory on the system, causing the reset of the session. The out-of-memory problem was resolved by utilizing the TextVectorization layer from the Keras library. Still, the problem was that the layer didn’t allow as much tuning as the vectorizer in sklearn. This might be due to the massive number of the data points that the system couldn’t handle.

2.2 Xgboost

2.2.1 Workflow of the source code

The workflow for Xgboost is the same as that of LDA except that feature selector (SelectKbest) is not used and new features of length and part of speech (POS) tagging are added to capture more information from the data in the pursuit of higher accuracy.

2.2.2 Learning Methods

Generally speaking, tree-based algorithms empower predictive models with high accuracy, stability and ease of interpretation. It does not require normalization or scaling of the data and maps nonlinear relationships well. XGBoost inherits the merits of the decision-tree-based ensemble algorithms and uses a superior gradient boosting framework that gives a combination of speed and performance optimization. It is a promising method to experiment on after the generative approach (LDA) and the neural network performing at an average level.

2.2.3 Feature Representations

The complexity of the English text usually depends on the choices of words and the length and grammatical structure of each sentence. For example, an independent clause with at least one subordinate clause is considered complex, because it is longer and contains several subjects, verbs and conjunctions. To better capture these features, in addition to vectorizing text with CountVectorizer or TfidfVectorizer, a new column of “length” is created by counting the number of words in each instance after lemmatization. A function of POS tagging is written to label each word based on word classes or lexical categories. Last but not the least, the additional dataset (dale_chall.txt) which contains about 3000 elementary English words is used to increase the weight of these simple words in the vectorizer.

2.2.4 Tuning parameters

The parameters affecting the accuracy score are max_features in the vectorizer and those in the Xgboost classifier such as n_estimators, max_depth, subsample, colsample_bytree, gamma and alpha. These parameters determine the structure of the tree and the complexity of the model to prevent overfitting.

2.2.5 Challenges

Xgboost is more complex than the linear models discussed earlier. Therefore, on average, it takes longer to fit one set of parameters than LDA. To run one iteration of model tuning through GridSearchCV with one value per parameter takes more than one hour using CPU. However, using the GPU provided by Google Colab reduces the time to around 15 minutes.

2.2.6 Evaluations

Different from LDA, as the number of max_features increases, the accuracy scores do not necessarily increase. Xgboost is not as sensitive to the max_features in the vectorizers as LDA. After tuning the parameters in Xgboost, the highest accuracy score is around 73% with the following parameters: max_features=10000, ngram_range=(1,3), min_df=6, max_df=0.85, learning_rate=0.01, colsample_bytree = 0.6, subsample = 0.8, objective='binary:logistic', n_estimators=1000, reg_alpha = 0.3,max_depth=35, and gamma=1.

Part B Unsupervised Learning

1. Motivation

Economic freedom is the ability of people in a society to select how to create, allocate, distribute, market different resources, while respecting others' rights to reciprocate. It is a term often associated with a government's economic, social and political policy. Rankings of economic freedom provides a comprehensive assessment of countries in the world. The goal of the project is to identify metrics that contribute significantly to the ranking and to find clusters of countries with similar characteristics based on these important metrics.

2. Data Source

- **Fraser Institute**
  Economic Freedom Rankings of 162 countries in the world from 1970 to 2018 can be downloaded in the excel format from the website. *(https://www.fraserinstitute.org/economic-freedom/dataset?geozone=world&year=2018&page=dataset&min-year=2&max-year=0&filter=0).* The web API allows customized download through filtering criterions such as countries, indicators and year. The dataset includes five major areas: 1)Size of Government, 2)Legal System and Security of Property Rights, 3)Sound Money, 4)Freedom to Trade Internationally and Regulation. Within the five major areas, there are 26 components in the index. Many of those components are made up of several sub-components. In total, the index comprises 44 distinct variables. Each row of the dataset represents a country. This analysis uses the data from 2018.

- **World Database**
  The dataset contains the basic demographics of the countries such as region, surface area, population, language, etc. The dataset can be downloaded in .sql format from the website*(https://dev.mysql.com/doc/index-other.html).*

3. Unsupervised Learning Methods

3.1 Data preparation

There are two steps in the data preparation process. The first step is to deal with missing values. Economic Freedom Summary Index which is the final score for each country is an average value of 5 major areas and sub-components. Based on this method of calculation, the missing values should be filled using the average value of either the major areas or sub-components. Second, 44 distinct variables from the year 2018 are retained, because the other variables are the average value of these distinct variables.

After the clustering is done, the labels from K-means clustering (k = 3) are merged to the World Database. Both datasets are merged on ISO Code columns.

3.2 Feature representations

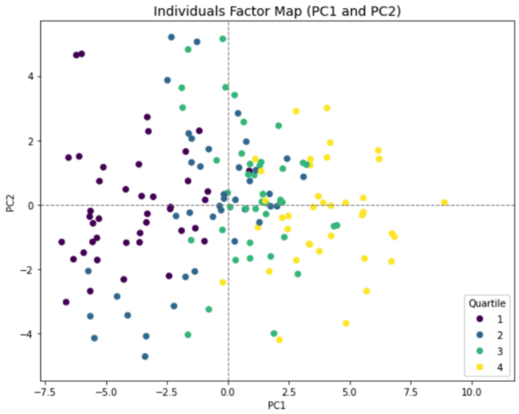
Principal Component Analysis (PCA) is a dimensionality-reduction method that transforms a large set of variables into a smaller one, while capturing most of the variance in the data. 44 different variables (dimensions) in the dataset pose a challenge in visualizing clusters in two dimension space as well as in generating meaningful clustering outcomes as correlation between variables introduces noise to clustering algorithms. PCA reduces the dimension of the data to 23 features which explains 90% of the variance. The first, second and third principle components which explains 28%, 9% and 7% of the data respectively construct a vector space to project individual data points for visualization and clustering analysis. Removing features with low variance acts as a filter that provides a more robust clustering. The directions with the most variance are usually most relevant to the clustering.

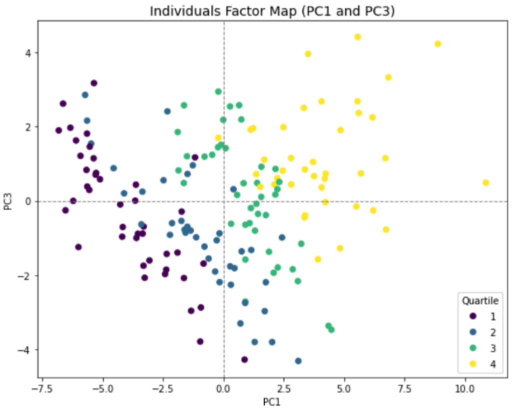
3.3 Learning methods

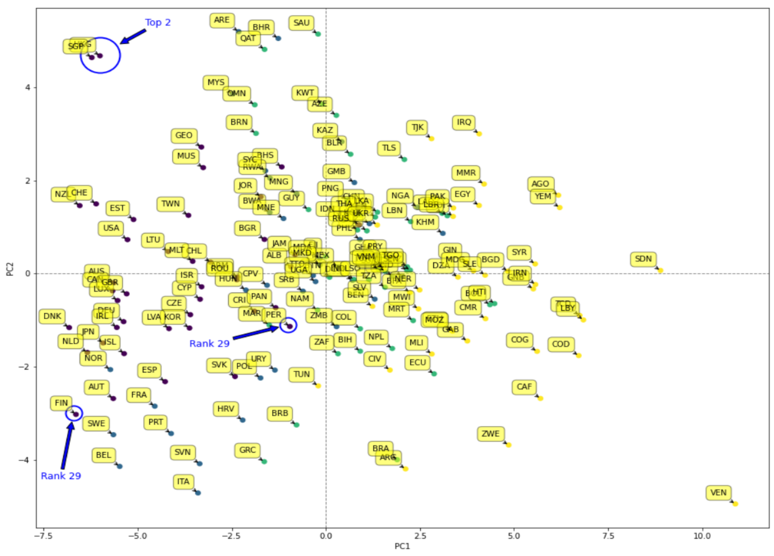
Visualizing all the data points in PCA constructed planes reveals that there are no significant density variations. Thus, to cluster countries based on similar characteristics, partition-based (K-means) and hierarchical clustering (Agglomerative Clustering) methods are chosen. Elbow method and silhouette analysis calculate the optimal number of clusters which is a predetermined parameter for K-means clustering. For agglomerative clustering, dendrogram, a tree-like diagram, illustrates the hierarchical relationship between objects. However, it does not determine the number of clusters and the “correct” number of clusters depends on interpretation. Therefore, finding the right number of clusters for agglomerative clustering can be challenging. Cross-referencing countries in different rankings and quartiles against different clustering results helps to determine the number of clusters that best represents the underlying patterns.

4. Evaluation

In the plane of PC1 and PC2, the majority of the countries in quartile 1 and 2 are positioned on the left side of the plane. HongKong (HKG) and Singapore(SGP) ranked number 1 and 2 respectively are positioned closely on the top left corner of the graph. However, countries like Finland (FIN) and Peru(PER) both ranked 29 are positioned far apart in the plane.The rest of the countries in quartile 3 and 4 are positioned on the right side of the plane.

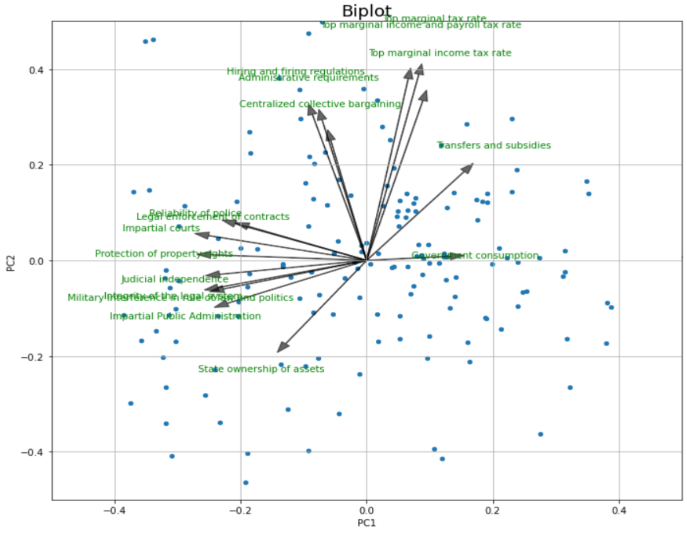






Below variable factor map suggests that the top 50% of the countries (left half of the graph) with higher economic freedom have higher value in protection of property rights, judicial independence, integrity of the legal system and impartial courts, which are essentially describing a government's ability to protect its people and reinforce justice and law without bias. These metrics summarize the representation of PC1.

PC2 is correlated to tax and regulation of business activities as well as labor markets. Interestingly, although countries ranked lower (right half of the graph) do not have trustworthy governments with sound legal systems, they tend to have higher value of index in government consumption and transfers and subsidies. It means these lower ranking countries have governments which spend less and are less likely to transfer wealth from some people to others. These in fact should have helped to boost economic freedom, because “when government spending increases relative to spending by individuals, households, and businesses, government decision-making is substituted for personal choice and economic freedom is reduced.” (Fraser Institute). Thus, the importance of having a trustworthy government with an impartial legal system outweighs the importance of the way a government distributes capital and conducts business.

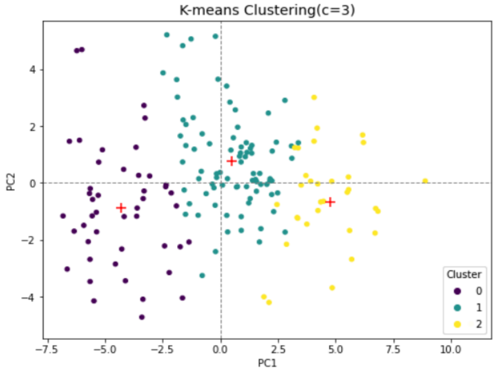


In the plane of PC1 and PC3, countries in the last quartile are mostly in the top right quadrant of the graph. These countries have unstable political status resulting in poor justice and legal system, less freedom of foreigners to visit and higher tariff rates, meaning that freedom of exchange goods and human capital across national boundaries is restricted. PC3 can be interpreted as the freedom to trade internationally and gender equality. K-means clustering is applied and visualized using PC1 and PC2. The Elbow method suggests that 2 clusters is optimal and it is in line with the corresponding factor map where the higher ranking countries are on the left half of the plane.

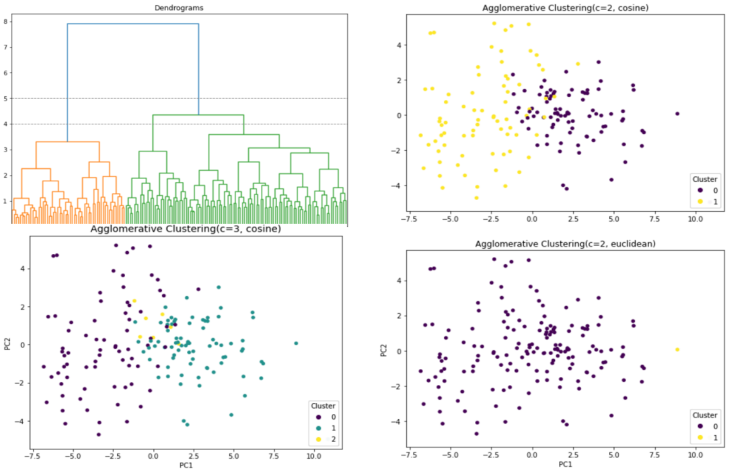
Silhouette analysis confirms that 2 clusters is the most optimal as all the silhouette scores are above average and the width of each cluster in the silhouette plot is similar. 4 clusters is a bad pick for the given data due to the presence of clusters with below average silhouette scores and also due to wide fluctuations in the size of the silhouette plots. However, the issue with 2 clusters is that it cannot support the analysis of the dissimilarities in terms of characteristics among countries ranked in the similar positions. For instance, both Finland and Peru rank 26th, but they are far apart in the plane of PC1 and PC2. Thus, clustering the countries into 3 groups may be the next best choice to get a deeper understanding of each cluster.

Based on the biplot (PC1 and PC2) and below graph (K-means, c=3):

- *Cluster 0* is made of countries such as HongKong, Singapore, US, UK, and Sweden. This group is characterized by strong protection of property rights, an independent and unbiased judiciary, and impartial and effective enforcement of the law
- *Cluster 1* is made of countries such as the United Arab Emirates, Qatar, Malaysia, Tunisia and Barbados. This group is characterized by low tax rate and relatively less restrictive for business entering into markets.
- *Cluster 2* is made of countries such as Brazil, Argentina, Sudan, Venezuela, Congo and Iraq. This group is characterized by rather unstable political status, inability to protect citizens' legal rights and to enforce justice and impartial laws. Government has little to spend due to the weak economy.



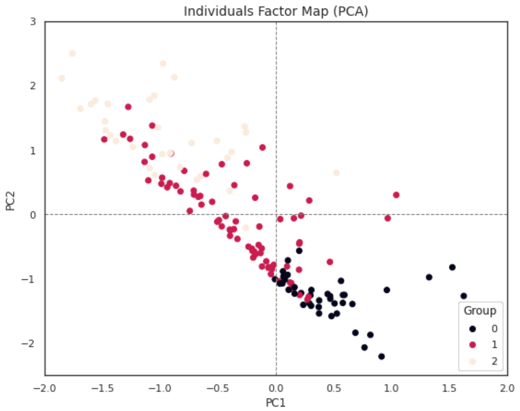
Using dendrograms to find the right number of clusters is subjective. As shown below, two horizontal lines indicate different thresholds. At the level of 5, there are two clusters and at the level of 4, there are 3 clusters. To determine whether these are the reasonable choices, agglomerative clustering with 2 and 3 clusters as well as cosine and euclidean metrics are plotted as per below. The result of agglomerative clustering with 2 clusters and cosine metric is much closer to the original quartiles than the other two. However, it inaccurately assigns certain countries such as Tajikistan (rank 132) to the higher ranking cluster. Agglomerative clustering does not seem to cluster as accurately as K-means, especially for those points near y axis (PC2) of the plane.

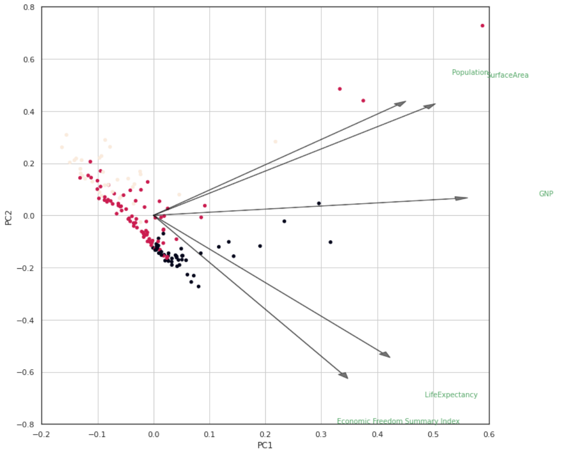


Using additional dataset to see if there's any demographic similarity between the countries in each cluster that is created from the previous k-means clustering (k = 3).
From the kde plots below, several things could be pointed out

- The distribution of Surface Area among the countries in each group is almost identical to each other.
- The distribution of Population and GNP among the countries in each group has roughly the same mean with each other but has a different variance.
- The distributions of the Economic Freedom Summary Index and Life Expectancy have a significant difference in means

Looking at the below Individuals Factor Map, the boundaries between each cluster are blurry, but they are still distinguishable from one another. Among four variables (Surface Area, Population, Life Expectancy, GNP), life expectancy was found to have an above medium correlation with the economic freedom summary index. According to Heritage Institute, economic freedom promotes improvements in the quality of health care, better access to clean water, better systems to remove waste, and better outcomes for AIDS and mortality incidence which lead to higher life expectancy (Patrick Tyrrell, Miguel Pontifis, 2019), which is consistent with the finding.





Discussion

1. **Part A:**
   There is no one-size-fits all approach to clean and vectorize text data. Rather, it depends on the dataset and the model chosen.It is often recommended to remove the noises such as the stopwords and non-alphanumeric characters and then lemmatize each

corpus to achieve a better prediction. However, it does not work in this case, because non-alphanumeric characters actually play an important role in differentiating classes and lemmatization decreases accuracy score.

Tuning hyper parameters and adjusting feature representation didn’t improve the result significantly. This was surprising because it didn’t fit the common notion of how feature engineering could greatly affect the model's performance. More time and resources could be used to find more effective feature representations.

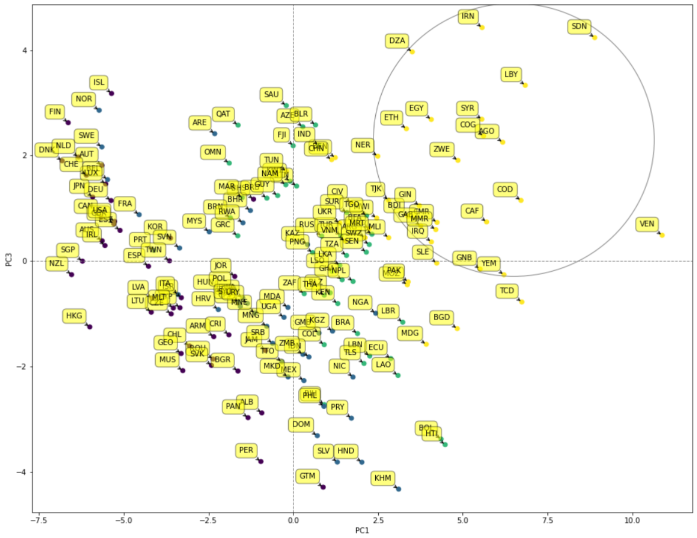
2. **Part B:**
   Clustering is a subjective process that requires some domain knowledge. Sometimes, it can be difficult to judge if the number of clusters found really makes sense. If more time and resources were given, it would be beneficial to apply the clusters found to downstream tasks such as classification or regression using supervised or semi-supervised learning techniques.
3. **Ethical Issues:**
   Possible selection bias could occur when labeling the data. For example, the standard on which text is difficult to understand might differ depending on the data labeler group's average academic achievement. This could potentially cause harm for those who have less ability to understand texts, such as children, people with reading disorders, or non-native English speakers. There are two ways to address this issue:
   a) Diversify labeler group using methods such as random sampling
   b) When choosing the metrics for the classification, put more weight on the recall score.

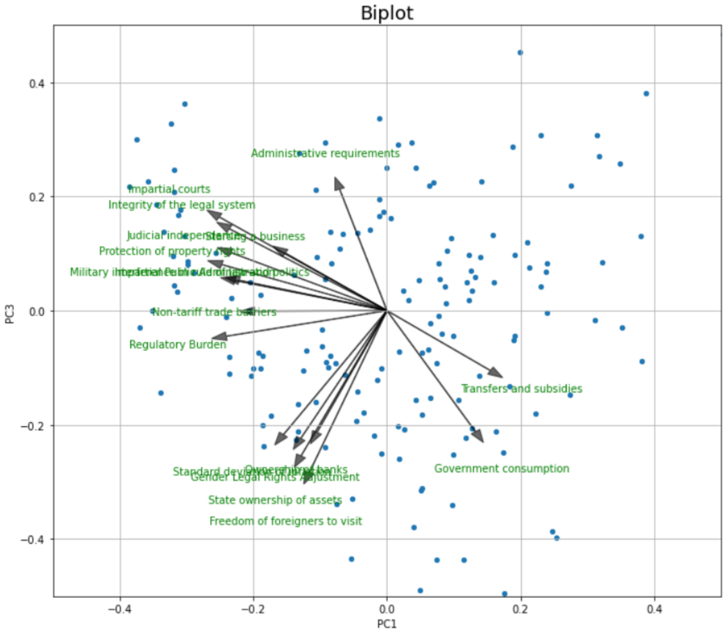
   Pre-existing biases are sometimes embedded in the data on which algorithms are trained. In the heritage dataset, missing values are ignored and the final score for each country is averaged only using the available data. This may inflate the scores of those countries with missing data and affect the overall ranking of economic freedom. Clustering based on such data might give some countries less accurate descriptions by placing them in the wrong clusters.

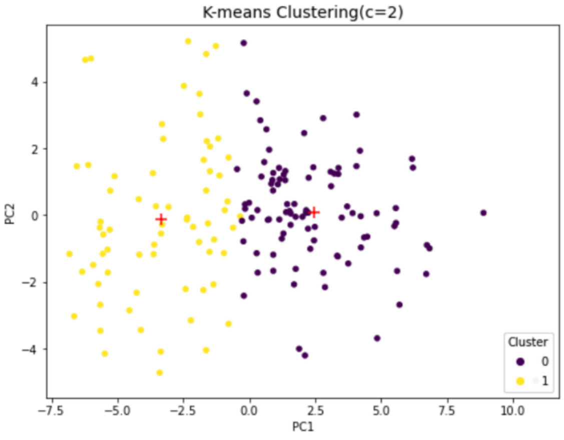
Statement of Work

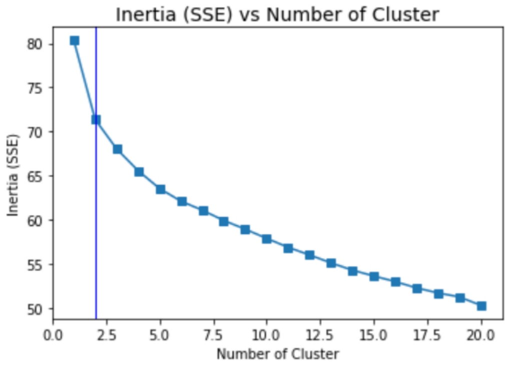
Sophie worked on the LDA and Xgboost model for supervised learning and applied PCA and clustering analysis in unsupervised learning. Do Young worked on the Bidirectional LSTM for supervised learning and utilized the additional dataset in unsupervised learning.

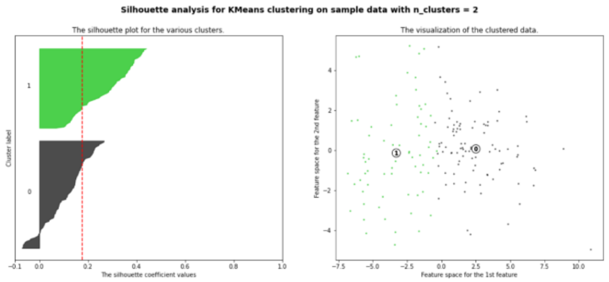
Appendix

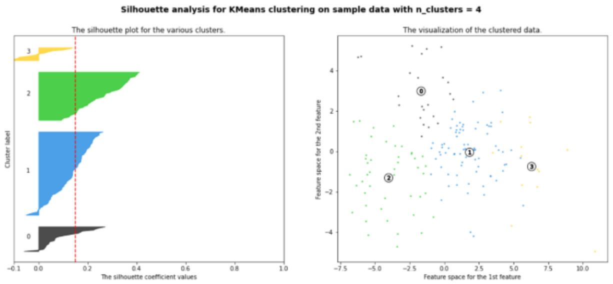

PC3
PC1
4
2
0
−2
−4
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
IRN
SDN
DZA
LBY
ISL
NOR
FIN
SWE
NLD
DNK
AUT
CHE
LUX
DEU
JPN
CAN
USA
FRA
AUS
IRL
SGP
NZL
HKG
SAU
QAT
ARE
BLR
IND
FJI
OMN
CHN
TUN
NAM
GUY
MAR
BHR
RWA
MYS
GRC
KOR
PRT
SVN
ESP
TWN
EGY
SYR
ETH
COG
AGO
NER
ZWE
TJK
GIN
CIV
SUR
UKR
TGO
RUS
VNM
MRT
MLI
IRQ
KAZ
PNG
TZA
SEN
LKA
SLE
GNB
YEM
COD
CAF
VEN
TCD
PAK
NPL
JOR
POL
HUN
LVA
ITA
LTU
MLT
HRV
URY
MNE
ZAF
THA
KEN
NGA
MDA
UGA
MNG
LBR
BGD
ARM
CRI
KGZ
BRA
COL
MDG
CHL
SRB
JAM
ZMB
LBN
ECU
TLS
GEO
SVK
TTO
MUS
BGR
MKD
MEX
NIC
LAO
ALB
PAN
PHL
PRY
DOM
BOL
HTI
PER
SLV
HND
GTM
KHM

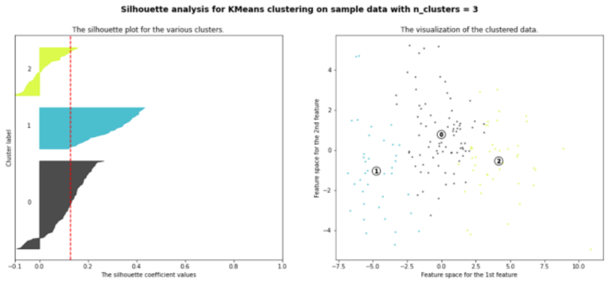

Biplot
PC3
PC1
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
Administrative requirements
Impartial courts
Integrity of the legal system
Judicial independence
Starting a business
Protection of property rights
Military interference in rule of law and politics
Non-tariff trade barriers
Regulatory Burden
Transfers and subsidies
Government consumption
Standard deviation of tariff rates
Ownership of banks
Gender Legal Rights Adjustment
State ownership of assets
Freedom of foreigners to visit

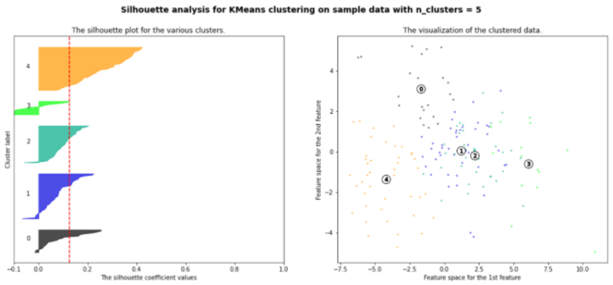

K-means Clustering(c=2)
PC2
PC1
4
2
0
−2
−4
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Cluster
● 0
● 1

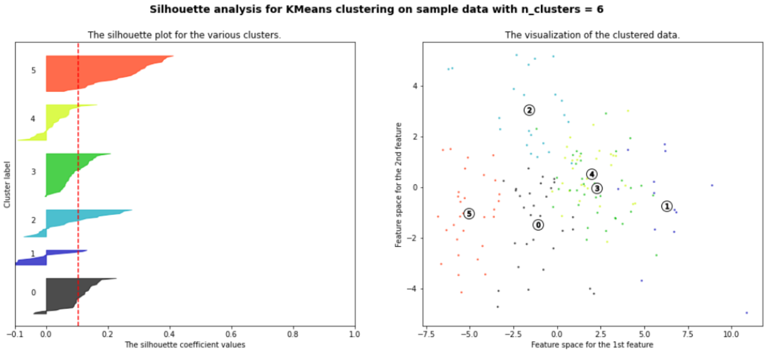

Inertia (SSE) vs Number of Cluster
Inertia (SSE)
Number of Cluster
80
75
70
65
60
55
50
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

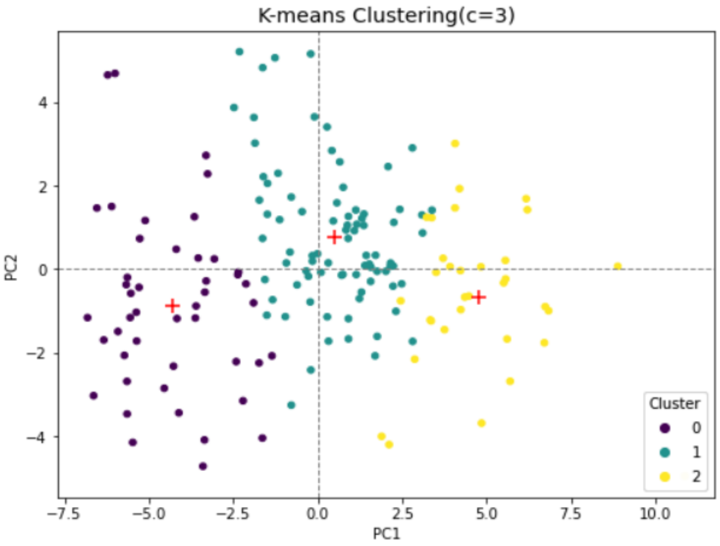

Silhouette analysis for KMeans clustering on sample data with n_clusters = 2
The silhouette plot for the various clusters.
Cluster label
1
0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature
0
1

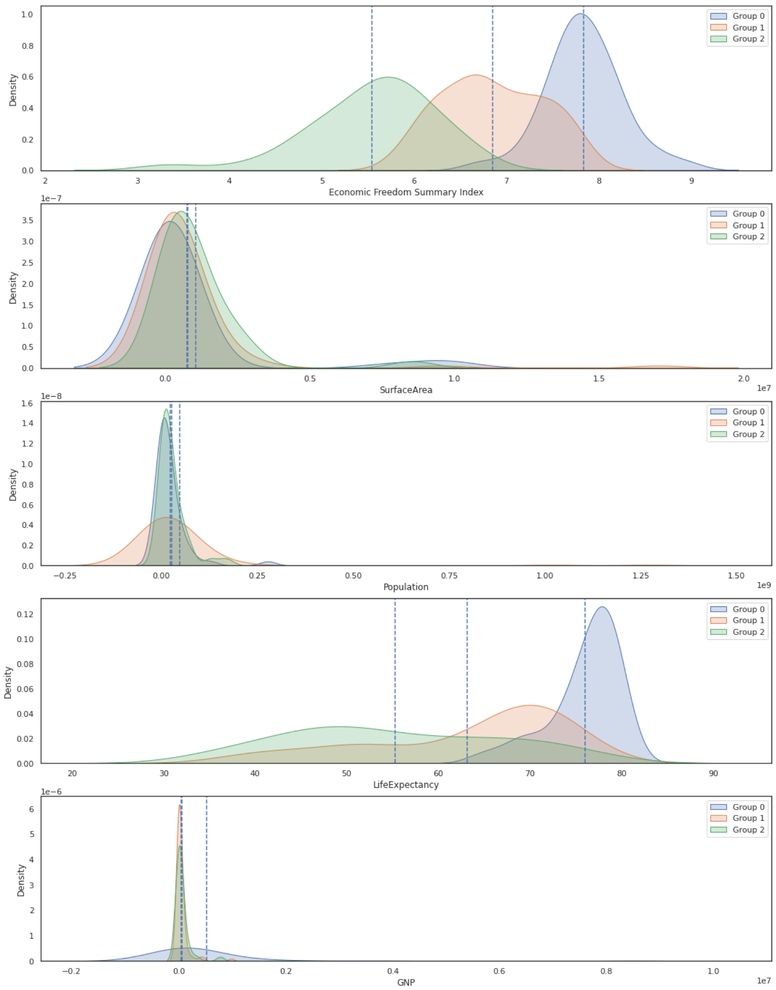

Silhouette analysis for KMeans clustering on sample data with n_clusters = 4
The silhouette plot for the various clusters.
Cluster label
3
2
1
0
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature
0
1
2
3


Silhouette analysis for KMeans clustering on sample data with n_clusters = 3
The silhouette plot for the various clusters.
Cluster label
2
1
0
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature
0
1
2


Silhouette analysis for KMeans clustering on sample data with n_clusters = 5
The silhouette plot for the various clusters.
Cluster label
4
3
2
1
0
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature
0
1
2
3
4


Silhouette analysis for KMeans clustering on sample data with n_clusters = 6
The silhouette plot for the various clusters.
Cluster label
5
4
3
2
1
0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0
The silhouette coefficient values
The visualization of the clustered data.
Feature space for the 2nd feature
Feature space for the 1st feature
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
4
2
0
−2
−4
0
1
2
3
4
5


K-means Clustering(c=3)
PC2
PC1
4
2
0
−2
−4
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Cluster
0
1
2


Density
Economic Freedom Summary Index
2
3
4
5
6
7
8
9
0.0
0.2
0.4
0.6
0.8
1.0
Group 0
Group 1
Group 2
1e−7
Density
SurfaceArea
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e7
Group 0
Group 1
Group 2
1e−8
Density
Population
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1e9
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Group 0
Group 1
Group 2
Density
LifeExpectancy
20
30
40
50
60
70
80
90
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Group 0
Group 1
Group 2
1e−6
Density
GNP
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1e7
0
1
2
3
4
5
6
Group 0
Group 1
Group 2