**Introduction**

This report aims to summarize the findings of a research conducted to see how different machine learning models predict whether a country would be classified as democratic or autocratic based on various independent variables. This study included 4 different machine learning techniques and two different sets of predictive variables to see which combination is most accurate in predicting the target variables. This paper contains 3 main headings. The first heading covers the general methods and procedure applied in the research along with detailed information about the dataset and the data cleaning procedure; the second heading covers the results and discussion where each possible model and predictive variable combination, the statistical outputs produced by them, and their interpretations are presented; and the last heading summarizes the entire paper along with possible limitations and future directions.

**Method & Procedure**

**Dataset**

The dataset was obtained from the BTI Transformation Index (<https://bti-project.org/en>), which is a “collaboration of nearly 300 country and regional experts from leading universities and think tanks worldwide.” Their aim is to collect data and lead transformational change in political and economic domains with their main purpose being steering the nation countries towards a more inclusive market and a democratic governance. Their team and methodology in collecting their data can be found on their website.

The dataset I created by merging other datasets is exclusively about the democratic, economic, and governing principles collected from nation states worldwide. Each domain is split into certain themes (for example, for democracy “Rule of law”, “Political participation”) and each theme is split into further themes (for example, for the “Rule of law” themes: “Separation of powers”, “Independent judiciary”, “Prosecution of office abuse”, “Civil rights”). Each higher-order theme score is calculated by averaging out the lower-order themes’ scores. Each domain also obtains their score the same way: by calculating the average of each themes’ scores.

In addition to these, each datapoint has a spatial and a temporal label attached to them. Spatial label is a regional label that consists of 7 different values which can be found in Table 1. The dataset I created by merging other datasets gathers data from 9 distinct datasets, all of which belonged to a separate year. By merging these datasets, I was able to obtain more datapoints and a temporal label. Temporal labels range from 2006 to 2022 with 2-year intervals in between.

***Table 1***

|  |  |
| --- | --- |
| **Code** | **Region** |
| 1 | East-Central and Southeast Europe |
| 2 | Latin America and the Caribbean |
| 3 | West and Central Africa |
| 4 | Middle East and North Africa |
| 5 | Southern and Eastern Africa |
| 6 | Post-Soviet Eurasia |
| 7 | Asia and Oceania |

**Data cleaning**

After merging the 9 different datasets into one, because there were over 110 variables, a new dataset was created where each column name had the number of missing values they contained corresponding to them. There were 2 obvious patterns among the missing values: i) For quite a lot of variables, there were exactly 73 missing values across the dataset suggesting that 73 datapoints were responsible for them and ii) certain variables had over 500 missing values out of nearly 1200 datapoints suggesting that either there were a technical issue caused during the merging process or certain variables could not be collected for almost half of the datapoints in the collection phase. Further checks confirmed all of the hypothesized reasons and i) the 73 rows were excluded from the dataset and ii) columns that had quite a lot of missing values due to technical issues (over 20 columns) and 4 columns that seems to have been consistently missing most likely due to the data collection problems were discarded from the dataset as well. This process yielded a dataset with no missing values so the remaining dataset was used for the analyses.

7 different variables were encoded using the LabelEncoder() because their values were of type string, just to have both versions in the dataset in case. These variables were: “Democracy/Autocracy”, “Status Index”, “Democracy Status,” “Economy Status”, “Governance Index”, “Governance Performance”, “Region”.

**Data analysis plans**

Two distinct sets of predictive variables were created to predict whether a country’s governance model was democratic or autocratic. The first set (General-Set) consisted of very broad or summarizing variables and the second set (Specific-Set) consisted of many variables each of which contributed to the summarizing or broad variables’ calculation to see which set would perform better and whether the more specific facets would yield overfitting or uncover very specific details such as a very specific facet contributing significantly to whether democracy or autocracy would prevail. All variables under the theme of democracy and governance were omitted since they were directly responsible for the target variable, and I wanted to see whether it would be possible to classify countries based solely on certain economic facets.

For each set of predictive variables, 4 machine learning techniques were utilized: K-nearest neighbor model, decision tree, logistic regression, and naïve Bayes classifier. In the following section, each model is run, and their relevant outputs are provided along with an interim discussion for both sets of predictive variables.

**Results & Discussion**

**General-Set**

***K-Nearest Neighbor***

Figure 1 shows the graph created to see which number of neighbors were most robust to run the model and the graph suggested it was 3.

***Figure 1***

A graph of a number of neighbors

Description automatically generated

However, the cross-fold validation process (GridSearchCV) imported from sklearn.model\_selection yielded 1 as the optimal number of neighbor number so both models were run to see how they would perform with the test data. The model with a single neighbor yielded an accuracy rate of 87.36% while it was higher for the model with 3 neighbors, 89.66%. The classification reports for the model with a single neighbor (Table 2) and three neighbors (Table 3) are presented below.

***Table 2***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.854 | 0.890 |  | 0.872 | 0.874 |
| recall | 0.865 | 0.881 |  | 0.873 | 0.874 |
| f1-score | 0.859 | 0.885 | 0.874 | 0.872 | 0.874 |
| support | 155 | 193 | 348 | 348 | 348 |

***Table 3***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.874 | 0.915 |  | 0.895 | 0.897 |
| recall | 0.897 | 0.896 |  | 0.897 | 0.897 |
| f1-score | 0.885 | 0.906 | 0.897 | 0.896 | 0.897 |
| support | 155 | 193 | 348 | 348 | 348 |

The model with 3 neighbors seems to have outperformed the model with a single neighbor in all respects. The 95% confidence interval calculated for the model with 3 neighbors with a 20 cross-fold validation process was 88.5% - 91.9% with a mean accuracy of 90.2%.

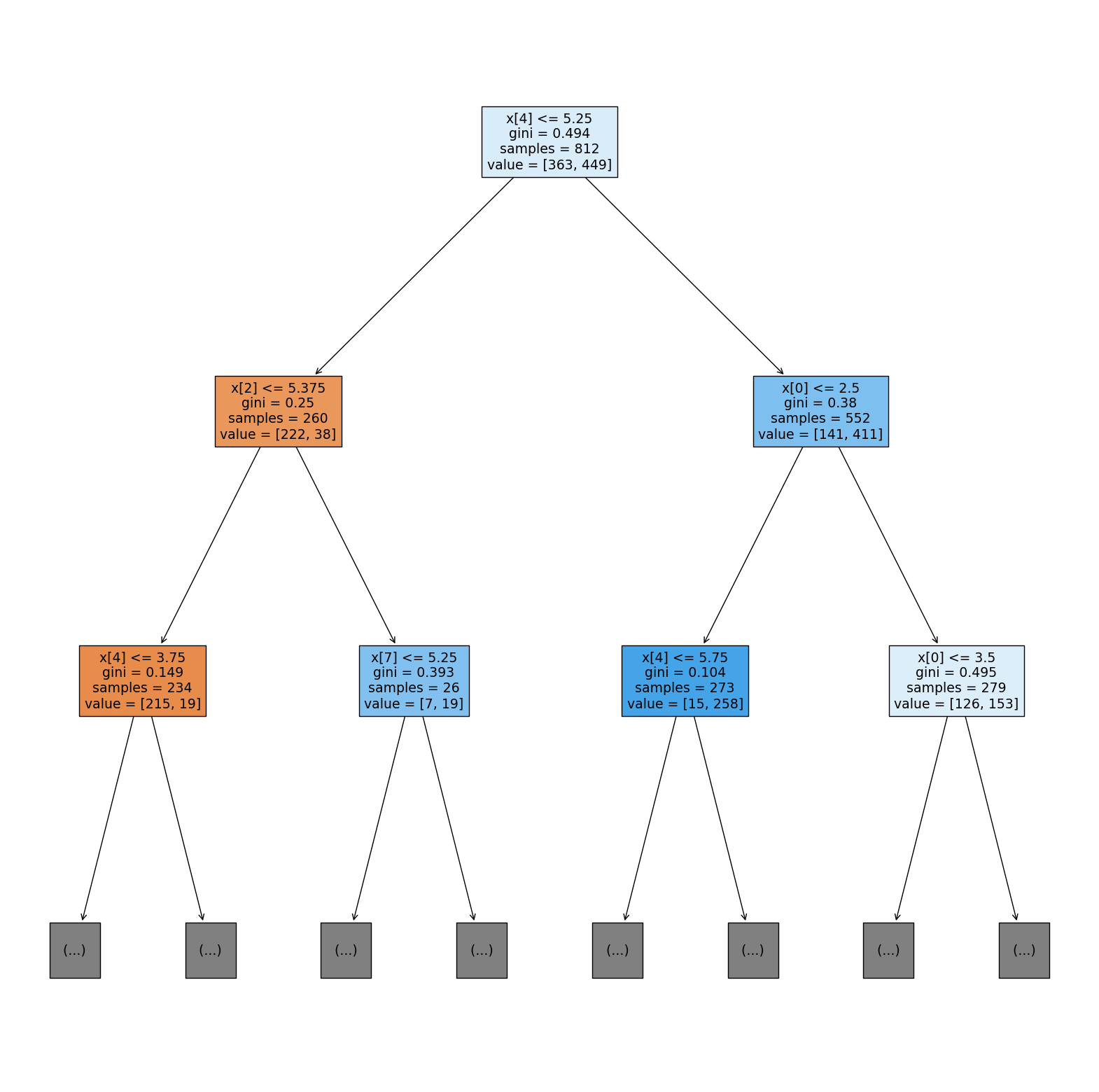
***Decision Tree***

The initial trial process that allowed the tree to branch out 25 folds yielded a model with 14 branches and a relatively good results overall for f1-scores etc. The optimization process decided the best parameters were gini for criterion, 7 for max depth, 18 for max features, and 3 for the minimum number of samples for a node creation. The classification report (Table 4), the first two branches for visualization purposes (Figure 2), and the visualization grid (Figure 3) are presented below:

***Table 4***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.807 | 0.904 |  | 0.855 | 0.861 |
| recall | 0.890 | 0.829 |  | 0.860 | 0.856 |
| f1-score | 0.847 | 0.865 | 0.856 | 0.856 | 0.857 |
| support | 155 | 193 | 348 | 348 | 348 |

***Figure 2***



***Figure 3***

A colorful lines and dots

Description automatically generated

The 95% confidence interval calculated for the decision tree model with a 20 cross-fold validation process was 84.7% - 88.1% with a mean accuracy of 86.4%.

***Logistic Regression***

Because it sometimes creates problems with logistic regression, the target variable was dummy coded as 0s and 1s and then the model was run. Ridge and Lasso penalizers did not result in significant changes (.02 differences at max) and there were relatively less predictive variables to create problems, so, the original model without any penalizers were kept. The classification report for the logistic regression model is presented below (Table 5).

***Table 5***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.775 | 0.771 |  | 0.773 | 0.773 |
| recall | 0.690 | 0.839 |  | 0.765 | 0.773 |
| f1-score | 0.730 | 0.804 | 0.773 | 0.767 | 0.771 |
| support | 155 | 193 | 348 | 348 | 348 |

The 95% confidence interval calculated for the logistic regression model with a 20 cross-fold validation process was 75.1% - 79.5% with a mean accuracy of 77.3%.

***Naïve Bayes Classifier***

Both Gaussian and multinominal models were used to see which would perform better. The classification reports for both Gaussian (Table 6) and multinominal (Table 7) are presented below

***Table 6***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.715 | 0.761 |  | 0.738 | 0.741 |
| recall | 0.697 | 0.777 |  | 0.737 | 0.741 |
| f1-score | 0.706 | 0.769 | 0.741 | 0.738 | 0.741 |
| support | 155 | 193 | 348 | 348 | 348 |

***Table 7***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.603 | 0.722 |  | 0.663 | 0.669 |
| recall | 0.697 | 0.632 |  | 0.664 | 0.661 |
| f1-score | 0.647 | 0.674 | 0.661 | 0.660 | 0.662 |
| support | 155 | 193 | 348 | 348 | 348 |

While the Gaussian model misclassified 90 datapoints out of 348 in the test phase, this was 118 for the multinominal model. The 95% confidence interval calculated for the Gaussian model with a 20 cross-fold validation process was 72.3% - 77.1% with a mean accuracy of 74.7% while the same interval for the multinominal model was 64.9% - 70.5% with a mean accuracy of 67.7%. The Gaussian model seems to have clearly outperformed the multinominal model.

**Specific-Set**

***K-Nearest Neighbor***

Figure 4 shows the graph created to see which number of neighbors were most robust to run the model and the graph suggested it was 1.

***Figure 4***

A graph of a number of neighbors

Description automatically generated

The cross-fold validation process (GridSearchCV) imported from sklearn.model\_selection also yielded 1 as the optimal number of neighbor number so the model with a single neighbor was set as the model unlike the process that ensued in the General-Set. The classification report for the model (Table 8) is presented below.

***Table 8***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.914 | 0.909 |  | 0.911 | 0.911 |
| recall | 0.885 | 0.933 |  | 0.909 | 0.911 |
| f1-score | 0.899 | 0.921 | 0.911 | 0.910 | 0.911 |
| support | 156 | 193 | 349 | 349 | 349 |

The model seems to have outperformed the k-nearest neighbor model trained with the General-Set. The 95% confidence interval calculated for the model with a 20 cross-fold validation process was 92.2% - 94.4% with a mean accuracy of 93.3%.

***Decision Tree***

The initial trial process that allowed the tree to branch out 25 folds yielded a model with 16 branches and a relatively good results overall for f1-scores etc. The optimization process decided the best parameters were gini for criterion, 11 for max depth, 19 for max features, and 4 for the minimum number of samples for a node creation. The tuned model’s classification statistics were poorer than the original model with a depth of 16 so I want with that model instead of the tuned version. The classification report (Table 9), the first two branches for visualization purposes (Figure 5), and the visualization grid (Figure 6) are presented below.

***Table 9***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.901 | 0.899 |  | 0.900 | 0.900 |
| recall | 0.872 | 0.922 |  | 0.897 | 0.900 |
| f1-score | 0.886 | 0.910 | 0.900 | 0.898 | 0.900 |
| support | 156 | 193 | 349 | 349 | 349 |

***Figure 5***

A diagram of a mathematical equation

Description automatically generated with medium confidence

***Figure 6***

A blue and red dots

Description automatically generated

The 95% confidence interval calculated for the decision tree model with a 20 cross-fold validation process was 88.2% - 92% with a mean accuracy of 90.1%. The decision tree model trained on the Specific-Set also seems to have outperformed the one that was trained with the General-Set.

***Logistic Regression***

The target variable was dummy coded as 0s and 1s again as it was the case with the first logistic regression model. Ridge and Lasso penalizers did not result in significant changes (.02 differences at max). The classification report for the logistic regression model is presented below (Table 10).

***Table 10***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.778 | 0.840 |  | 0.809 | 0.812 |
| recall | 0.808 | 0.813 |  | 0.811 | 0.811 |
| f1-score | 0.792 | 0.826 | 0.811 | 0.809 | 0.811 |
| support | 156 | 193 | 349 | 349 | 349 |

The 95% confidence interval calculated for the logistic regression model with a 20 cross-fold validation process was 80.2% - 83.2% with a mean accuracy of 81.7%. Like the other two models, logistic regression trained with the Specific-Set seems to have outperformed the logistic regression model trained with the General-Set.

***Naïve Bayes Classifier***

Both Gaussian and multinominal models were used to see which would perform better. The Gaussian model outperformed the multinominal in this case again. The classification report is presented below (Table 11).

***Table 11***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Aut. | Dem. | accuracy | macro avg | weighted avg |
| precision | 0.704 | 0.751 |  | 0.728 | 0.730 |
| recall | 0.686 | 0.767 |  | 0.726 | 0.731 |
| f1-score | 0.695 | 0.759 | 0.731 | 0.727 | 0.730 |
| support | 156 | 193 | 349 | 349 | 349 |

The Gaussian model misclassified 94 datapoints out of 349 in the test phase, which is 4 more datapoints compared to the model trained with the General-Set. The 95% confidence interval calculated for the Gaussian model with a 20 cross-fold validation process was 71.7% - 76.7% with a mean accuracy of 74.2%. The model trained with the General-Set and Specific-Set have overlapping confidence intervals but nevertheless, the mean accuracy score and the classification statistics for the model trained with the General-Set are slightly better.

**Discussion**

The average of macro f1-scores and %95 confidence intervals are presented in Table 12.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | K-Nearest | | Decision Tree | | Logistic Regression | | Naïve Bayes | |
|  | GS | SS | GS | SS | GS | SS | GS | SS |
| Macro avg for f1-score | 0.896 | 0.91 | 0.856 | 0.898 | 0.767 | 0.809 | 0.738 | 0.727 |
| 95% CIs | 88.5%  -  91.9% | 92.2%  -  94.4% | 84.7%  -  88.1% | 88.2%  -  92% | 75.1%  -  79.5% | 80.2%  -  83.2% | 72.3%  -  77.1% | 71.7%  -  76.7% |

***Table 12***

*Note: GS: General-Set, SS: Specific-Set, CI: Confidence interval*

The best performing model both according to macro average f1-score and 95% confidence intervals is the k-nearest neighbors model trained with the Specific-Set. The k-nearest neighbor model is also the best performing model regardless of the dataset used while the worst performing is naïve Bayes classification model, again regardless of the training dataset used. Out of 4 models, the Specific-Set performed better with 3 of the models while the General-Set performed better in the Naïve Bayes classification model. The better performance of the Specific-Set might be because it did not have that many variables (15) that would have led to a noise or an underfitting/overfitting problem because the dataset had enough datapoints and variation to allow for more variables and more precision without having much of a problem. Nevertheless, the best performing model and dataset combination is the Specific-Set with k-nearest neighbors model with a 93.3% accuracy rate on average and a 0.91 macro average for the f1-scores. These results suggests that with proper datasets and machine learning algorithms, we could categorize countries into autocratic and democratic regimes based solely on economic factors and potentially screen for or forecast the likelihood of a country’s shift towards greater democracy or autocracy, or calculate thresholds, minimum requirements etc. for a stable democracy based on economic factors, holding other relevant variables constant.

**Conclusion**

This brief research aimed to see whether it was possible to develop a machine learning algorithm to classify countries as democratic or autocratic based on economic factors. I tried out a total of 8 different combinations with 4 different machine learning models (k-nearest neighbors, decision tree, logistic regression, and naïve Bayes classification) and 2 datasets (one with more variables one with less). The k-nearest neighbors model with the more detailed dataset performed best out of all possible combinations with a macro average f1-score of .91 and a mean accuracy rate of 93.3% across a 20 cross-fold validation scheme.

It is important however to point out to the inherent limitations of this research study. First of all, this is a correlational study at best as no causal analysis or causal theoretical framework to support the relationship between the target and predictive variables were present. Although there were no implicit or explicit assumptions about economic factors leading to or causing democracy or autocracy, this should be explicitly emphasized: It might be the case that economic factors might steer a country towards a certain type of regime, or democratic or autocratic practices leading to certain economic policies, or a bidirectional relationship between the variables that creates a feedback loop. However, even if the results are correlational in nature, this should not prevent one to use such models to forecast or ‘screen for’ possible shifts from democracy to autocracy or vice versa. Another limitation of this study was that the dataset was already collected and as far as I could tell, the dataset was biased towards a neoliberal understanding of economy, which might create a lot of potential problems. Furthermore, other relevant variables that might have led to a greater precision (such as a social domain with gender equality, treatment of minorities etc. as being the themes) were not present in the dataset. Keeping these limitations in mind, the reader is advised to interpret and apply the results of this research with caution.