Technical Report

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
THE GLOBAL GOALS For Sustainable Development

The Global Goal #4: QUALITY EDUCATION

A toolkit for researchers and decisionmakers around the globe

Matti Kiviluoto, Juhani Kivimäki, Sebastian Lampinen

20.10.2020

Contents

Introduction

We discussed several options for the project, but soon found common ground in the United Nations (UN) Sustainable Development Goals (SDG). One goal resonated exceptionally well among the three of us – #4 Quality Education. We decided to investigate the data gathered by UN to assess how well countries meet the goals. But that was just one data source and we wanted to couple it with another extensive source. After some pondering, we agreed on the CIA Factbook as it was an interesting juxtaposition with regards to the SDG data. Luckily, after some trouble with data wrangling, we found out that the two data sources augmented each other in a nice way and opened opportunities to build models to map basic country level statistics with the SDG #4 indicators. Suddenly, we saw that this model could be used in decision-making to evaluate the impact of one country level statistic on an SDG #4 indicator. Equally importantly we saw that with our clustering and modelling researchers could investigate the effects of a country level statistic on a specific SDG #4 indicator. Whoever is interested can fairly easily extend the model to other SDG goals as well – provided by the same API – and thus find even more opportunities to use our work to support decision-making or to conduct research.

Data Collection and Preprocessing

Our data was collected from two sources. From the UN SDG Data Hub website https://www.sdg.org/#api and from Ian Coleman's website https://iancoleman.io/exploring-the-cia-world-factbook – a third-party API to the CIA Factbook data. Both sources provide a staggering amount of data as .json files.

Although the UN SDG API provides a clear interface to the data, the problem of grasping which indicators were available remained to be investigated. Luckily, this was solved by a listing via the API which we converted into a multi-workbook EXCEL-file, which groups the indicators with the same grouping as in the data.

From the available SDG data, we chose *Indicator 4.4.1: Proportion of youth and adults with information and communications technology (ICT) skills by sex and type of skill (percent)* to be used for our predictions. Some preprocessing had to be made since the `.geojson` file contained missing values. We chose to take the means of values of each country for simplicity. We built our models in order that they work with different SDG data that is preprocessed similarly.

The CIA data came in a preprocessed form. Main problem was, how to find relevant data that was relevant and usable in building our regression model. The nested structure of the `factbook.json` file meant a lot of manual digging to find the usable information. A set of 33 variables was handpicked as the set of independent variables to be used in the regression. These were chosen based on two principles. First, data was to have potential in making the predictions. Second, data was to be comparable. As an example of the first point, the highest point in any given country probably will not reveal much about its success in attaining SD goals. A second case example could be different ethnic and religious groups in any given country, which are very specific to that country, and cannot be compared in a meaningful way across multiple countries.

Next, a code-based interface for reading data from the `factbook.json` was created. This was implemented in the class `Reader`, which is inside the file `reader.py`. The constructor takes a `.json` file as a parameter and creates a `Reader` object. The `Reader` object implements a `read_data()` method, which takes as a parameter a list of query words defined by user (by uncommenting from a ready-made list). Each query word matches an independent variable. The interrelation between query words and variables is given in the file `CIA_factbook_manual.xlsx`. The `Reader` object takes care of the backend processes needed to find the information asked by the user from the nested structure of the `factbook.json` file. If some information is missing, a `NaN` value for that variable is inserted instead. The `Reader` object also implements a `get_missing_data()` method, which the user can use to check what data is not available for a given country.

Main functionality is implemented inside the file `CIA_SDG_regressor.py`. The main method starts by opening the `factbook.json` and passing it to `Reader` constructor. Next, the list of user chosen query words is passed to the `read_data()` method of the `Reader` object just created, which returns the asked information in a pandas `DataFrame` with countries as indexes and variables as columns. Then, geographic area codes are added as a column to this `DataFrame`. These are manually inserted inside the file `country_codes.csv` and are used to match the CIA data with the SDG data. Next, user is given an option to manually drop some regions from the analysis. These can be larger regions (such as World or EU) or outliers found in the exploratory phase. Finally, countries with any `NaN` values are dropped.

The SDG-data and the CIA-data did not have a joint key as for some reason the CIA data did not use any standard code for the countries, luckily spelling of the names of the countries were fairly similar with each other. We could determine a linkage between the ISO-coded SDG data and the CIA data and implemented a method that combines the datasets. Here we discovered two unfortunate countries, where politics made matching impossible – Taiwan and Palestine (Taiwan not listed in the UN data and Palestine not in the CIA data). Sadly, we had to drop these countries from our data.

Exploratory data analysis

Before building the machine learning model, some exploratory analysis was executed. A quick discovery was that choosing some variables led to a situation, where a large number of countries were dropped. Some consideration was given to whether imputation of missing values was to be implemented. However, this was ruled out mostly due to the fact, that the dataset already contained a rather small number of data points. Imputing missing values might have led to a poorer performance of the regression model. Rather, the choice of variables was left to user discretion.

Some tools for exploratory analysis were implemented. The method `transform()` uses `PowerTransformer` from the `sklearn.preprocessing` package to standardize the values of the independent variables. This method was chosen because most of the variables were not normally distributed. These standardized values are then passed to method `principal()`, which does three things. First, it performs k-means clustering for the variables, with the number of clusters given by user in the method call. Then, it performs dimensionality reduction with principal component analysis (PCA) to several components given by the user in the method call. Finally, it creates a scatter plot with first principal component on the x-axis, second principal component on the y-axis and third principal component as point size. The clusters are indicated by the different colors of the points. At first, PCA was performed to non-standardized variables, which led to some large-scale variables (notably population size and area) dominating the variance. After standardization, results became more useful.

Machine Learning

Our goal was to predict the SDG data values for countries that were missing in the SDG data using the CIA data. First, we built a Linear Regression model to predict these values. We evaluated the model with 3-fold cross validation with R-squared as the scoring metric. The means of cross validation scores were around 0.3 when choosing reasonable variables from the CIA-data to use as the predictors.

Since Linear Regression predicted negative values for some dependent variables, and negative values were off the scale, a non-linear regression model was devised. We decided to use a Random Forest Regressor as our second take on the problem. First, the question of how to choose optimal hyperparameters for the model, given that the choice depended on what variables the user had decided to use, was raised. For this problem, a hyperparameter optimizer was implemented in the method `rfr_optimizer()`. It takes the independent and dependent variables as parameters, constructs a random grid out of different handpicked hyperparameter value combinations. These values were carefully chosen after thorough testing to always provide for a feasible set of hyperparameters, no matter what variables the user chooses to use. The method makes use of `RandomizedSearchCV()` method provided by `sklearn.model_selection` package. It goes through a preset number of iterations (100 in this case), chooses a random set of hyperparameters from the random grid on each iteration, and performs a 3-fold cross validation to evaluate the set of hyperparameters. Finally, it returns the set of best hyperparameters found.

After optimization, a Random Forest Regression model is constructed with the given hyperparameters. This model is evaluated once more with 3-fold cross validation with R-squared as the scoring metric. With a reasonable selection of independent variables, the mean of cross validation scores ranged from

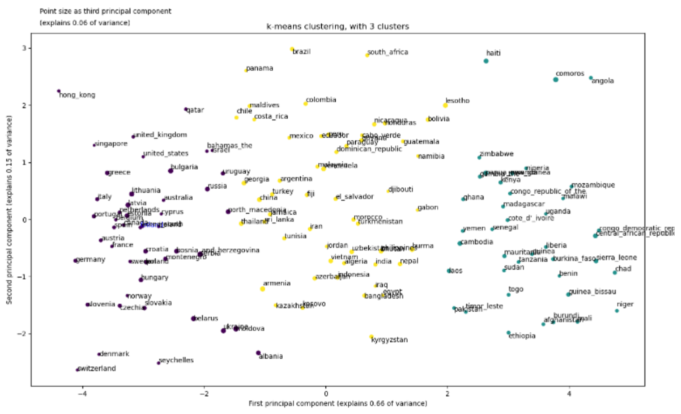
0.74 to 0.77 which is a significant improvement over the Linear Regression model. Finally, the regression model is used to predict the missing values for the SDG data, just as with the Linear Regression model.

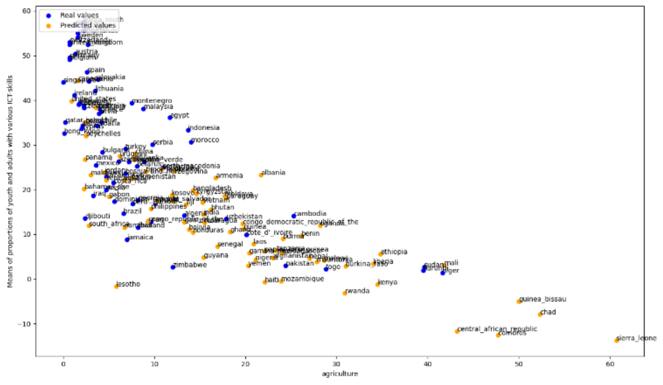
Visualizations

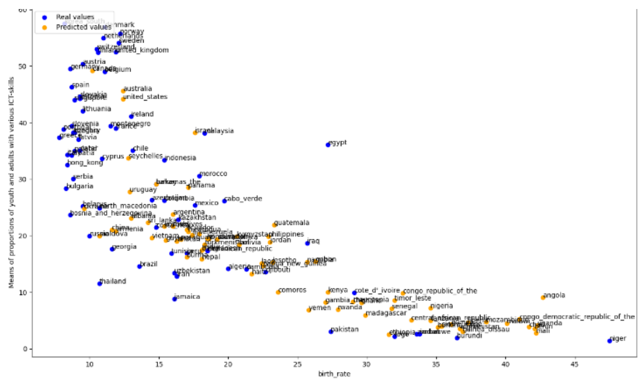
We created scatterplots of the real and the predicted SDG data values as y with each of the chosen explanatory variable from CIA data as x. This helps the user see which variables of the CIA data correlate with the SDG data. When using only variables with clear correlation with the SDG data, we found that the predictions were more accurate.

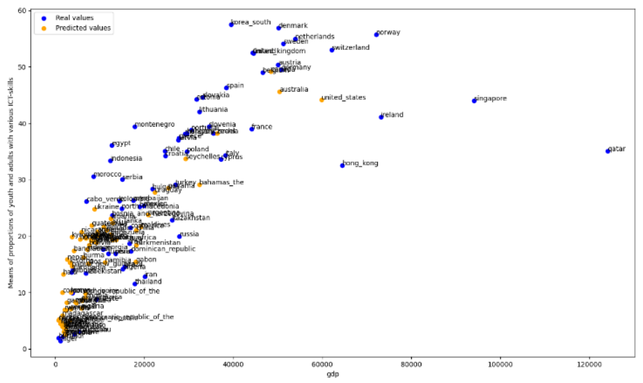
One added benefit of these scatterplots was, that outlier points were easy to detect. If a given country was clearly separated from the other data points in several different scatter plots, it was classed as an outlier. Manually removing these outliers from the training set, led to significant improvements in the model accuracy. This is something the user must do on a case by case basis.

Some sample scatterplots:









Communication of Results

This is the full list of features we provide for our users:

- Full human readable listing of all SDG4 indicators available from the UN SDG API as well as a field listing of the CIA factbook data.
- Preprocessed CIA factbook background data from countries across the globe.
- A model to use a country's CIA data to predict values in the UN SDG data.

- Option for the user to choose both the explanatory variables from the CIA data and the SDG data to use as the target of linear regression or random forest regression.
- Tools for exploratory data analysis are provided, including principal component Analysis and k-means clustering with user defined number of dimensions and clusters.
- Further options to narrow down the list of countries to analyze.
- Scatterplots of the real and predicted SDG values as y with each of the chosen explanatory variable as x for the chosen variables.
- These clear visualizations make it easy to detect outlier data points. These outliers can be excluded when building the regression model by user's discretion.

Further points of interest

The next steps to augment our solutions would be to add prepared data from the other SDGs. To make our models easier to utilize for users without an understanding of Python would mean a need to create a user interface to our models, where users would be presented by options and multiple-choice fields to choose the different fields to be chosen.

The GitHub-site

We have gathered all our data and models to our Github-site for anyone interested to use.
See: https://github.com/mattikiviluoto/SDG4
To augment the executable files, we have added descriptive files that describe the fields of our two data sources as well as a presentation to explain what our package contains. On the Github-site you can find a wiki that offers help to users on how to use and edit our models to fit their purposes and to allow future users to build on our work to extend it even further.

Conclusion

This has been an interesting deep dive into the world of data science. We have been able to see the pain points actual data scientists encounter and gotten a glimpse of where the hard work lies when wrangling data and building meaningful models. Although expected, the work needed to get workable data and to see the shortcomings it had were a surprise, but in our case, they inspired us to work to overcome these shortcomings. The ease of using machine learning models (once all the hard work is done) was a satisfying moment. With someone else having provided the nice libraries, all we had to do was to try them out and compare the different models. We also enjoyed the roleplay of acting as researchers for policymakers – as if we were discovering novel approaches to build a better world.