SAPIENZA
UNIVERSITÀ DI ROMA

Country Happiness Predictor

Computer Science Master Degree
Machine Learning Exam

Niccolò Morabito
Matricola 1808746

Prof.ssa Paola Velardi

A.A. 2020-2021

Contents

1 Introduction **3**

2 Data collection **4**

3 Pre-processing and feature engineering **5**

3.1 Feature selection and extraction 5

3.2 Feature transformation 6

3.2.1 Missing values 6

3.2.2 Categorical features 6

3.2.3 Normalization 7

3.2.4 Target feature 7

3.2.5 Imbalanced classes 8

4 Training **9**

4.1 Model selection 9

4.2 Results 10

5 Conclusions **12**

1 Introduction

The idea of this project was born reading and studying the World Happiness Reports [1] [2], that each year show the estimated happiness of a country. The results are based on respondent ratings about their lives in their own country. Therefore, one of the great criticisms of this research is that it appears subjective and it cannot be taken into account by governments because it does not provide objective reasons behind certain results.

The aim of this document is to show if the WHR is actually reliable by studying the underlying characteristics of countries, in order to understand if there is a correlation between objective and measurable features and the happiness score which is computed interviewing people.

Should such a relationship be confirmed and should a Country Happiness Predictor be trained, then it would be simpler to study and understand respondent ratings. Moreover, it could also be possible to see the impact that a political choice would have on the happiness of the citizens. For instance, if a country wants to discourage births, it could be estimated how a decrease in the fertility rate could affect the overall happiness of the people. The same study could be done on the social issues, e.g. monitoring unemployment, labour force participation by sex and education enrollment.

2 Data collection

The main problem of studies about countries is the lack of data. The number of countries in the world is 195 [3]; if we consider also dependencies or other territories, this number can increase up to 234, however, clearly there cannot exist a dataset with a greater number of instances unless different years are taken into account [see Section 5].

The total number of instances is even smaller in this case of study, since the World Happiness Report collects its data only in a little more than 150 countries.

The WHR dataset contains six features in addition to the actual happiness score. These factors constitute elaboration of a lot of data in order to make the countries easily comparable [4], but in this way they lose specificity and objectivity. Therefore, it is important to get another dataset to collect different data from the same set of countries. In particular, UNData dataset [5] contains key statistical indicators of the countries which can be considered representative:

- general information;
- economic indicators;
- social indicators;
- environmental indicators.

In both cases, even if there are happiness reports for different years, 2017 has been taken into account since the countries data of UNData dataset refer to this year.

The first part of the work consists thus in *inner*-merging the two different datasets on the "Country" column, which is common to both ones, and pre-process the data in such a way that the objective indicators and features can be used to train a machine learning model for predicting the happiness score.

Of course, the bottleneck is the WHR dataset: if a country does not have any happiness score, the corresponding row is going to be discarded from the UNData dataset too. Moreover, the following countries are involved in the World Happiness Report but no data of them was contained in the UNData dataset: Ivory Coast, Kosovo, North Cyprus and Taiwan.

More information about datasets composition and pre-processing process to lose the smallest amount of information possible can be found in the next section [see section 3].

3 Pre-processing and feature engineering

The most important part of this work has been to pre-process the available data and to analyze features in order to extract and aggregate the relevant information without losing discriminatory factors nor overfitting the system. As previously anticipated, the first task was to merge the two datasets such that for each country in the UNData dataset there was a corresponding happiness score. In order to do that, some manual checks have been necessary since countries were sometimes named in different ways (e.g. "State of Palestine" instead of "Palestinian Territories" or "Syrian Arab Republic" for "Syria").

3.1 Feature selection and extraction

Looking at the dataset resulting from the merging process, it is possible to notice that some columns must be dropped off. In particular, it turned out that the WHR dataset should maintain only the *Happiness score*, which is the metric measured by asking the sampled people the question: "How would you rate your happiness on a scale of 0 to 10 where 10 is the happiest?" [4] and that will be used as the target feature. *GDP per capita* is duplicated, the *Happiness rank* is useless since it is relative to other rows; finally, the remaining columns (e.g. *Whisker*, *Family*, *Freedom*, *Generosity*...) were non-objective for the reasons stated above.

Some columns should be removed also from the UNData dataset, since they are not useful for training (e.g. *Region*, [see subsection 3.2.2]) or due to ambiguous data (e.g. *International migrant stock* or *Net Official Development Assist. received*).

Moreover, some features from the UNData dataset contain two values separated by a slash, that correspond to the two sexes, two ranges or two geographical distinctions. In all these cases, two columns have been produced first, and in some cases, eventually aggregated [see next subsection].

Column name	Number of NaN values
Education: Primary gross enrol. ratio (f per 100 pop.)	6
Education: Primary gross enrol. ratio (m per 100 pop.)	10
Education: Secondary gross enrol. ratio (f per 100 pop.)	10
Education: Secondary gross enrol. ratio (m per 100 pop.)	18
Education: Tertiary gross enrol. ratio (f per 100 pop.)	15
Education: Tertiary gross enrol. ratio (m per 100 pop.)	21
Education: Government expenditure (% of GDP)	21

Table 1: Columns with more missing values.

3.2 Feature transformation

3.2.1 Missing values

The majority of columns with missing data has < 5 `NaN` values, therefore the best solution is to impute them from the other instances. A `KNNImputer` from `sklearn` library has been used, setting `n_neighbors=10` and `weights="uniform"`.

The exceptions are the columns with data about the educational aspect as summarized in the Table 1. About the first six columns, it was already planned to aggregate the data contained in them. It is still necessary to impute missing values before, but the overall mean is going to be less affected by these fictitious values. The first attempt was to maintain the same division of other two-values columns, ending up with two columns: *Education: average gross enrol. ratio (f per 100 pop.)* and *Education: average gross enrol. ratio (m per 100 pop.)*. In this way, also the sex ratio was taken into account. However, it has been seen that in this way all the observed metrics (accuracy, average precision, average recall and average f1 score) suffered about 5% loss. That is why the six columns were aggregated into one single column called *Education: average gross enrol. ratio (per 100 pop.)*.

About the last column named *Education: Government expenditure (% of GDP)*, it was decided to exclude it since having 14% of missing values - even if this could mean to lose important data. Public investment in education is probably the most important thing to affect school enrollment, therefore the two features could be considered related and to some extent redundant.

Region	Continent
Caribbean	North-America
CentralAmerica	South-America
CentralAsia	Asia
EasternAfrica	Africa
EasternAsia	Asia
EasternEurope	Europe
MiddleAfrica	Africa
NorthernAfrica	Africa
NorthernAmerica	North-America
NorthernEurope	Europe

Region	Continent
Oceania	Oceania
South-easternAsia	Asia
SouthAmerica	South-America
SouthernAfrica	Africa
SouthernAsia	Asia
SouthernEurope	Europe
WesternAfrica	Africa
WesternAsia	Asia
WesternEurope	Europe

Table 2: Correspondence between region and continent in the combination which led to the best performance.

3.2.2 Categorical features

After working on the two-values columns, the only features that turned out to be non-numeric are *Country* and *Region*, which contain strings. The former constitutes the primary key of the

dataset, therefore it can easily be removed taking for granted that the name of a country is not going to affect the happiness of its inhabitants. About the latter, it was initially hypothesized that a geographical limitation could help to represent the living conditions of the population. That is why the *Region* column was initially taken into account for training the model.

Obviously, since it is a categorical feature, it needs some kind of encoding into numeric values. Using label encoding - each possible value of the categorical feature is simply converted to a number - some models (like Decision Tree) would lead to having non-sense splits, and then one-hot encoding is the only alternative.

The values of the *Region* column are: "SouthernAfrica", "EasternAfrica", "South-easternAsia", "SouthernAsia", "WesternAsia", "NorthernEurope", "SouthAmerica", "SouthernEurope", "MiddleAfrica", "CentralAmerica", "Caribbean", "Oceania", "WesternEurope", "CentralAsia", "WesternAfrica", "NorthernAfrica", "EasternAsia", "NorthernAmerica" and "EasternEurope", then one-hot encoding produces 19 more columns. Considering that the total number of features after all the pre-processing explained in the previous sections is 53, 19 additional columns is a big value and it constitutes almost one-third of the total number of features.

As could be expected after the last observation, this approach leads to worse performance with respect to results obtained removing *Region* column, most likely because it only represents additional noise.

However, there is still another attempt that can be carried out believing in the importance of the geographical area of the country: transforming the Region into the corresponding Continent. Since there are several ways of distinguishing the continents [6], different combinations have been tried (e.g. distinguishing between North and South America slightly increased overall performance). The final choice is summarized in the Table 2. Nevertheless, the best combination does not reach either the performance of the same model trained without the *Region* feature, that is why the column was eventually removed.

3.2.3 Normalization

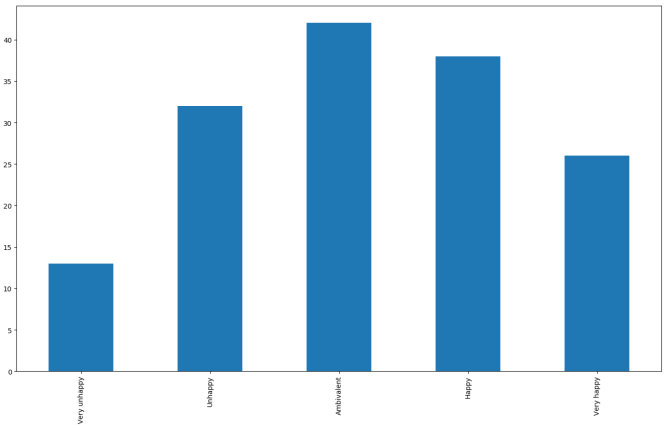
As explained in the subsection 4.1, a neural network is going to be trained with this pre-processed dataset. Therefore, it will also be necessary to scale the input features before providing them to the input layer.

3.2.4 Target feature

As stated in the section 3, the target feature is *Happiness score* from the World Happiness Report dataset. Its values are continuous, therefore the predictor should be a regressor; however, in order

to deal with the low number of instances in the dataset and simplify the problem, the target feature has been discretized into five classes which represent respectively very unhappy, unhappy, ambivalent, happy and very happy countries.

Of course, this simplification is going to affect the usability and the usefulness of the predictor, but it was seen that a greater number of classes strongly affected the performance because of the irrelevant number of instances per class. By augmenting training data it could be possible to increment the number of target classes and, probably, to output a continuous value, as explained in the section 5.



Target class	# of instances
Very unhappy	13
Unhappy	32
Ambivalent	42
Happy	38
Very happy	26

Figure 1: Classes imbalance.

3.2.5 Imbalanced classes

After target feature discretization, the distribution of instances through the classes is summarized in the Figure 1. As expected, the classes are not balanced and the resulting distribution is almost normal. In order to address this problem, it was decided to oversample the dataset using **Synthetic Minority Oversampling Technique (SMOTE)**. This is obviously going to increase performance since we are replicating the available instances without using an effective data augmentation technique, as will be explained in the subsection 4.2 and the section 5. But it has proved necessary also in order to address the low number of instances in the dataset.

Finally, the total number of instances increased to $42 \times 5 = 210$, instead of the initial 151.

4 Training

4.1 Model selection

The previous sections should have clarified that this work focused on pre-processing and feature engineering more than hyper-parameters tuning or algorithm selection. It could not be otherwise considering the lack of data previously nominated, that prevented carrying out a full and reliable hyper-parameters tuning. Splitting the dataset into training and test sets considering a ratio of about 2:1, means to have $210 * 70\% = 147$ instances in the training set, that is $147 \div 5 \approx 29$ examples for class. And only after adding 59 instances with SMOTE [see subsection 3.2.5].

Splitting the training set further with - for instance - a 3-fold cross-validation approach would have mean expecting a model to predict a certain class after training it with only 10 examples for that class. Nevertheless, a validation set has been extracted from the total data. In particular, the test set was composed of one-quarter of all the instances and the remaining ones were further split into training and validation sets with a 3:1 ratio. Of course, the overall number of instances should not be considered sufficient for such a division, but instead, it constitutes just an attempt for didactic purposes. The section 5 will draw the appropriate conclusions about this choice.

Originally, the validation set was intended for early stopping the neural network training before overfitting. In fact, neural networks tend to continuously increase the performance until reaching the maximum but, after a certain amount of epochs, what they learn is no more useful nor generalizable, and a validation set is important to recognize this behaviour through a decrease in performance on it. In the end, it was also used to perform some hyper-parameters tuning and to select the best model not only within the ones with different HPs but also between different algorithms.

In particular, three different algorithms were used: Random Forest and Decision Tree (with `RandomForestClassifier` and `DecisionTreeClassifier` of `sklearn` respectively) and Neural Network, implemented with `pytorch`. About the first two, the hyper-parameters tuning phase was carried out using `GridSearchCV` to test different combinations of parameters, selected from the most common and most promising ones according to the scientific literature. For the neural network, instead, a simpler version was manually implemented to tune only its most important hyper-parameters. The network has been designed with a variable number of linear hidden layers which are activated by the same activation function; the optimizer uses stochastic gradient descent and has a learning rate of 0.01 (bigger values prevent the network to converge); finally, the criterion used by the loss function measures the mean squared error between each element in the input and

the target. The tuneable parameters are the number and the size of hidden layers, the activation function and the so-called "early stop patience", which represents the number of times in which the performance on the validation set decreases that are waited before stopping the training process. Of course, a reduction in validation performance does not necessarily mean that the model has started to overfit, that is why this last hyper-parameter is going to test different values to understand if a greater value allows improving the model.

Algorithm	Best f1-score
Random Forest	83.71%
Decision Tree	82,71%
Neural network	82,43%

Table 3: Best results achieved on validation set by the different algorithms.

4.2 Results

The metric that has been used for selecting the best model is the f1-score, which has been averaged on the different classes. In the Table 3 with the best f1 scores for each algorithm, it is possible to notice that the best random forest model has the greatest performance, even if there is little difference with the others. That is why the final evaluation phase with the test set has been performed on that model, and the results are summarized in the report of the Table 4, which distinguishes the results according to the different classes, and the Table 5 that provides the scores for the most important metrics.

	Precision	Recall	F1-score	Support
Very unhappy	1.00	0.89	0.94	9
Unhappy	0.64	0.88	0.74	8
Ambivalent	0.75	0.55	0.63	11
Happy	0.83	0.71	0.77	14
Very happy	0.79	1.00	0.88	11

Table 4: Classification report for best random forest model on test set.

The average accuracy is 79%, which is similar to the other considered metrics. The average f1-score is 79% too, even if some classes (e.g. "Very unhappy" or "Very happy") seem to be predicted correctly more easily. The first thing that must be noticed, however, is that the data on which this evaluation is performed is not enough, as already widely explained. There are about 10 instances for each class, and the results cannot be considered really reliable for this reason. Moreover, we also have to take into account that synthetic oversampling has been carried out to address classes imbalance, therefore the real result could be less positive. Oversampling only in the training and validation sets would have obviously helped to have more reliable (and probably less satisfactory)

results, but this led to having too few instances in the test set, especially for some classes that should have been tested with less than the current and already-insufficient 10 instances.

Accuracy		**Precision**	**Recall**	**F1-score**	**Support**
0.79	Average	0.80	0.80	0.79	53
	Weighted average	0.80	0.79	0.79	53

Table 5: Final metrics for best random forest model on test set.

Nevertheless, some first good results have been achieved and all tested models acted pretty well after a simple hyper-parameters tuning. Even the neural network, which seemed the weakest approach considering the lack of data, was able to learn, showing good results although it sometimes easily started to overfit after a low number of epochs. Luckily, the tuning of the hyper-parameters helped to prevent this behaviour using the early-stopping technique explained above with the right value.

5 Conclusions

This project should be considered as a first attempt to predict happiness of a country from other objective information. It does not claim to be fully reliable nor to correctly predict the happiness score of a country, and it does not prove that such a predictor could be as accurate as the World Happiness Report.

The aim of this study was to show that there is a strong correlation between the economic and social indicators of a country and the happiness of its inhabitants. On one side, such relationship could give more credibility to the statistical surveys that the United Nations Sustainable Development Solutions Network (SDSN) carry out each year. On the other hand, it could really help countries to focus on the right factors to pursue the happiness of its citizens.

The obtained results are promising, but further work should be carried out to understand if a more reliable and usable model is possible. In particular, the most urgent attempt to make is augmenting the dataset, because this would help from different points of view: having more data could obviously lead to a more generic and accurate model; moreover, it could be possible to avoid oversampling to address classes imbalance; also, the evaluation part could be more reliable; finally, this would contribute to increase the number of the output classes or even move to a regression problem with a consequent increase in the precision and utility of the predictor. In order to do so, data from different years could be involved in the training process. Since 2016, every year (on the 20th of March, i.e. the UN's International Day of Happiness) the new World Happiness Report is published [2]. Therefore, we only need to collect the countries data from 2016 to 2021 to have about five times the current number of instances that has been used for this project. Moreover, taking into account a longer period for training a model would also reduce the possible bias of the predictor. One thinks of the past year: if the training data was totally taken from 2020, the results were probably going to be inaccurate due to the unusual situation which would have (and has) changed the citizens' priorities.

References

[1] *The World Happiness Report.* URL: https://worldhappiness.report/.

[2] *The World Happiness Report, Wikipedia.* URL: https://en.wikipedia.org/wiki/World_Happiness_Report/.

[3] *How many countries are there in the world?* URL: https://www.worldometers.info/geography/how-many-countries-are-there-in-the-world/.

[4] *FAQ — The World Happiness Report.* URL: https://worldhappiness.report/faq/.

[5] *UNdata, a world of information.* URL: http://data.un.org/Host.aspx?Content=About.

[6] *Continent, Wikipedia.* URL: https://en.wikipedia.org/wiki/Continent.