Noa Eker – 311399984; Avi Caciularu – 203056585; Yossi Tamarov – 203036256

EDUCATION AND ECONOMIC FACTORS AS HAPPINESS PREDICTORS

PROJECT OVERVIEW:

Is it possible to explain a man's happiness using educational and macro-economic parameters of the country he lives in? This is the question we chose to answer via the prediction of a chosen happiness index, the Happy Planet Index, using the databases provided by the World Data Bank, and specifically Edstast dataset. To interpret the HPI using educational and macro-economic factors, we used Random Forest algorithm, which produced a prediction with precision level of $R^2 = 0.718$. Among the relevant parameters that were used for the prediction, one can mention the following: Gross enrolment ratio, primary, both sexes (%); Official entrance age to pre-primary education (years). Running the model takes roughly an hour.

NOTEBOOK OVERVIEW:

In order to check whether our prediction is strongly based upon the countries and years provided in the dataset, in addition to the original dataset (hereinafter will be referred to as 'main data') we created two more copies of the basic dataset, the first is countryless, i.e. is without the country feature (hereinafter will be referred to as 'no countries'), and the second is yearless, i.e. is without the year feature (hereinafter will be referred to as 'no years'). The copying takes place in cell 3.7. In order to watch the process on each dataset, we used widgets e.g. radio buttons and multiple choice tabs to allow the reader to choose between 'main data', 'no countries' and 'no years'.

In order to shorten the notebook's running time, we implemented saving and loading of long-time-generated graphics, so for generating the graphics from scratch – please click on the button 'Delete offline plots'. For further explanation about the buttons see 1.1.

REQUIRED INSTALLATIONS (USING ANACONDA PROMPT & PIP):

1. Make sure conda is up to date, and if not, please run: `conda update conda`
2. Seaborn module: `conda install -c anaconda seaborn=0.7.1`
3. Scikit-learn module: `conda install scikit-learn`
4. Geonamescache module: `sudo pip install geonamescache`
5. Basemap module: `conda install -c conda-forge basemap=1.0.8.dev0`

0. INTRODUCTION:

BUSINESS LOGIC:

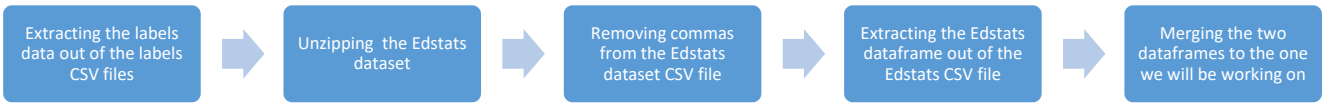
The macro-economic language, in the form in which it is common both in the economic media and in among the decision makers in the Israeli government, bases its happiness measurement on economic growth. Hence, as the state's product increases, meaning the sum of the services and products it supplies in a certain year, so does the happiness in the state increases. The assumptions that this approach is based on are: The more a man consumes and owns, the merrier he gets and that men are insatiable, and thus know no limit for their desire to consume and own. Out of a profound disagreement with these assumptions, thinking that in our era humanity got to understand the negative implications of an excess use of the resources our planet can provide us, and the importance people give to an equal and fair distribution of wealth, we ask to measure happiness using a new index, the Happy Planet Index (hereinafter will be referred to as HPI). This Index is composed of Gallup's international wellbeing surveys, life expectancy in each country, the variance of these two factors and finally on the ecological footprint of each country. The HPI is preferable over economic growth also thanks to its ability to measure subjective wellbeing, as it is perceived by each individual, but at the same time it is its weakness, since it is the very factor we want to influence and don't know how to. Therefore, we chose to use different data to explain the HPI, so it can be used for policy making.

1. DATA PREPARATION & CLEANING :

DATASET PREPARING (CELLS 1.1- 1.3):

Importing utilities.py file which contains classes and code that we use in our data-science project (cell 1.1). The purpose of this file is encapsulation and taking care of the visibility and good-looking of the notebook. Utilities.py uses 3 more files which it imports: classes.py which contains the significant part of the code which the notebook uses, global_variables.py which saves mostly important paths for directories and files of our project, and imports.py which runs import statements. Running this cell will also trigger option buttons – 'Run long executions' for enabling long runs, i.e. data preparation run and full model training for each model, 'Don't show plots' to disable plots and visualizations, and 'Delete offline plots' to clear saved graphics in order to regenerate them along the run of the notebook.

The data preparation which takes place in cell 1.2 (when clicking the 'Run long execution' button) is explained in the next flow chart: (the label's datasets were obtained from the Happy Planet Index site)



LOADING THE DATASET FILE (CELL 1.3):

The Edstast dataset was initially chosen because of its fitting features – macro-economic and educational data, which can be seen in this cell, e.g. Adjusted net enrolment rate, lower secondary, both sexes (%).

NULL VALUES HANDLING (CELL 1.4):

Columns and rows with $\%(nulls) > 80\%$ were dropped. This specific threshold led to best results, since other thresholds perform poorly, e.g. 90% left us with almost no features. After performing this stage, 67 features are left in our dataset (out of 3624 features at start).

2. DATA UNDERSTANDING:

BASIC DATA ANALYSIS (CELLS 2.1-2.3) CONCLUSIONS:

- The most prominent countries in our datasets are Western countries.
- In recent years (2005+), the amount of data is significantly higher.
- The HPI column values appear to be normally distributed among our dataset.

CORRELATIONS (CELLS 2.4, 2.5):

We used Spearman's correlation, which only assesses monotonic relationships (whether linear or not). In the correlations matrix plot, we included the 5 most correlated features to HPI (determined in cell 2.4). In the plot, one can see all the Spearman's correlation coefficients between 6 columns: our chosen 5 features and the HPI (our label). It is easy to infer that parts of our features are highly correlated (for example, different population measures), which might lead to collinearity and overfitting, which we will deal later on.

DATA PLOTS ON MAP (CELL 2.6):

This plot enables to check visually whether the most correlative feature to HPI ('Population, ages 15-64, female') is "enough" in order to roughly predict the average HPI per country. Looking at the graph, the answer in no - there isn't enough similarity between the true average HPI per country and the average of the correlative feature per country.

DETECTING DATA SKEWNESS (CELL 2.7):

We plotted boxplots in order to study the distributional characteristics of our features. As you can see in the example in the cell, data is not symmetric (mean is far from median).

3. PREPROCESSING:

IMPUTATION TECHNIQUE (CELL 3.1):

Remaining null values were imputed by median. We chose median since, observing boxplots (e.g. cell 2.7), one can deduce the data exhibits some skewness (using other imputation techniques, like mean value, resulted in higher error percentage).

DATA TYPES (CELLS 3.2, 3.6):

We used One Hot Encoding technique to transform the categorical field ‘country’ to a binary format that works better with regression algorithms, and numeric values were converted to floats.

SPLITTING AND PARTITIONING DATA (CELLS 3.3-3.4, 3.7):

In these stages, we randomly split our training data to train and test. In addition, we created copies of our data without the year and country features as mentioned above. As said before, if accuracy does not decrease significantly, then the new datasets are considered "better" than ‘main data’ (including the country and year columns). Otherwise, we should prefer ‘main data’.

BINNING BY DECADES TECHNIQUE (CELL 3.5):

This technique was used on the ‘year’ field, based on other Kaggle notebooks and based on comparison to other binning techniques.

LINEARITY PROVING (CELL 3.8):

In this section, we checked the main linear regression assumptions on the train data. These assumptions are required in the outlier detection part. We used:

1. Residuals vs. predicted plot: The random pattern here suggests that a simple linear model is appropriate.
2. QQ-plot of residuals: the relationship between the theoretical percentiles and the sample percentiles is approximately linear (although light tailed) which suggests that the error term is normally distributed, as required.

Applying OLS on train data, the obtained R^2 is 0.86, which implies the regression line approximates the real data points well.

SCALING NUMERIC FEATURES (CELL 3.9):

We performed feature standardization: $\frac{x-\bar{x}}{sd}$, (i.e. making the values of each feature in the data have zero-mean and unit-variance). This method is widely used for normalization in statistical learning. Later, we used Elastic net regularization which also requires standardization.

4. OUTLIERS DETECTION:

1. ROBUST REGRESSION METHOD (CELL 4.1):

Since the linear regression assumptions are met (already proved), we utilized robust regression method to detect outliers. The idea behind the following algorithm is that applying robust linear regression on data results in significantly high residuals for extreme observations, and thus allowing us to remove outliers according to the magnitude of their residuals. Of all loss functions, we chose the Huber loss that is, as required, less sensitive to outliers in data than the squared error loss. This specific loss function was picked since the obtained regression line fitted the data well. Other options we tried included: RANSAC loss and TheilSen loss. The Huber regressor differs from TheilSen regressor and RANSAC regressor because it does not ignore the effect of the outliers but gives a lesser weight to them. Consequently, it enables detecting outliers efficiently.

The outliers removal algorithm works as follows:

1. Robust regression is applied on train data (Huber regressor)
2. The model residuals average and standard deviation are computed.
3. Observations whose residuals' absolute values are far from the mean by at least 2 standard deviations are removed.

Applying the above algorithm on our train data, 44 rows are dropped. The validation R^2 (*) is improved from 0.91 to 0.95 and the residuals vs. predicted plot gets a more random pattern, as desired.

(*) The validation R^2 is a statistic obtained by averaging the OLS R^2 on 100 random subsets of a given dataset (in our case, the dataset is our train data).

2. PCA (cells 4.2.1-4.2.3):

Applying PCA on our data and applying K-Means clustering on the result of the PCA in order to detect interesting centroids (cell 4.2.1). Eventually, we managed to detect high leverage points (which can be easily observed in the graph) which belong to India and China (cell 4.2.2). Removing those points from our dataset, we get overall accuracy difference in the prediction that is not higher than 0.03, and thus choose to leave the outliers in the training set (cell 4.2.3).

5. Feature selection with elastic-net model

Feature selection (cells 5.1-5.3):

The feature selection is done by Elastic-Net (which is a regularized regression method that linearly combines the $l1$ and $l2$ penalties of the Lasso and Ridge methods). Like the Lasso method, Elastic-Net pushes to zero features which are not "significant", thus can be used as a feature selection method. However, when there are collinearities among variables, Lasso selects "randomly" one variable and zeros the coefficients of the rest. In our case, multicollinearity exists. Unlike the Lasso, the Elastic-Net tends to keep groups of correlated predictors in the model. Thus, the obtained model is more stable.

ELASTIC-NET RESULTS:

- 89 features are left after elastic-net feature selection (out of 222).
- 3 most influential features (excluding countries) are: Official entrance age to primary education (negative coefficient), year (positive coefficient), and annual population growth (negative coefficient).

In Addition, comparing the prediction based on the features that are selected by the Elastic-Net and the prediction based on the features that the Lasso regression as a feature selection method chooses, we saw that the later provides poor results, e.g. $R^2 = 0.485$.

6. Feature extraction

Synthesizing new features (cell 6.1):

We extended our linear model to a second-degrees polynomial (using a greater degree caused overfitting and high variance).

CHECKING CORRELATIONS BETWEEN TARGET AND SYNTHESIZED FEATURES (CELL 6.2):

Afterwards, we computed Spearman's correlation coefficient between the new quadratic terms and HPI (our label). It turns out that there are 460 quadratic features that have higher correlation coefficient than all of the original features. One can deduce these quadratic terms should be included in our model.

7. KERNEL RIDGE REGRESSION

Motivated by previous conclusion, we chose polynomial (2nd degree) Kernel Ridge regression as our main model (our model's hyperparameters were chosen using grid search).

8.MODEL EVALUATION

COMPARISON BETWEEN LINEAR REGRESSION, RIDGE REGRESSION AND RANDOM FOREST (CELLS 8.1-8.3):

The 'linear' models (OLS and Ridge) were added relying on the linearity prove in cell 3.8.

Random Forest was added here since this model is widely used (and as non-linear comparison model).

Ridge regression: As expected, results are quite similar to linear regression results (Since features were already chosen by Elastic-Net, which is close to Ridge regression).

Random Forest: This model appears to be the best one out of the 4, concerning all the measures we examined. In all the mentioned models, the prediction distribution on test data is more centered around the mean (less variance). Looking at the plots we created for each model (see appendix for all images combined) we can conclude that the Random Forest model overfits the training dataset (see error percentage per year and per GDP). It can also be seen that this model predicts the HPI the best over both GDPs and years (see mean prediction on test dataset per year and per GDP). Finally, it seems that the Kernel Ridge model predicts the distribution of the HPI in the most accurate way.

RESULTS (CELL 8.4):

The table compares the 4 models, according to R^2, mean HPI and Error percentage (*) (on test data).

Among the 4 models: Random Forest preforms the best, Linear regression's and Ridge regression's performance are quite similar and poor and Kernel Ridge regression's performance is slightly better than OLS and Ridge, yet inferior to Random Forest.

(*) Error percentage is computed as follows:

1. For each 'observation' in our test dataset, the deviation percentage of the predicted label from the true label is computed.

2. The mean deviation percentage (as computed above) of all observations is called the 'Error percentage'.

9. PREDICTION OF RANDOM FOREST

The two plots here enable to check visually, per country, whether the Random Forest mean HPI prediction is close enough to the true mean HPI.

Comparing the graphs, one can deduce they are mostly similar, thus the prediction is good enough.

10. TYPES OF DATASETS PREDICTION COMPARISON

The table presents comparison between our original dataset, i.e. ‘main data’, the 'no years' dataset and the ‘no countries' dataset, concerning 7 measures.

- Two of the measures, R^2 and error percentage are directly related to accuracy of prediction.

1. The 'no countries' dataset's R^2 is negative, therefore the dataset is bad (we rather simply predict any sample as equal to the mean). The error percentage is very high, as expected.

2. The 'no years' dataset's R^2 is significantly lower than the one of ‘main data’.

- The mean prediction for test dataset differs among the 3 datasets, and is closest to the true mean in ‘main data’.

To conclude, both 'no countries' and 'no years' datasets aren't enough for reasonable prediction.

11. CONCLUSIONS

We conclude that the Happy Planet Index, though based on subjective wellbeing can be predicted by totally objective factors that the policy makers in each country can achieve, consisting of educational factors and macro-economic factors. Moreover, since it can be seen in the mean HPI per GDP plots that HPI is not correlated with GDP, it seems that this index has its own insights about happiness, which are not captured correctly by GDP, which is the regular index for growth measurement and hence for happiness in macro-economic policy making. Thus, we suggest using the Happy Planet Index as a measure for happiness, and our model shows that it can be used for policy making with familiar parameters, while taking into account both ecological aspects and inequality aspects.

REFERENCES

Why and how to use Elastic-Net regression: http://stats.stackexchange.com/questions/184029/what-is-elastic-net-regularization-and-how-does-it-solve-the-drawbacks-of-ridge

World Data Bank dataset: *EdStats: Education Statistics*, http://datatopics.worldbank.org/education/

Happy Planet Index site: http://happyplanetindex.org/

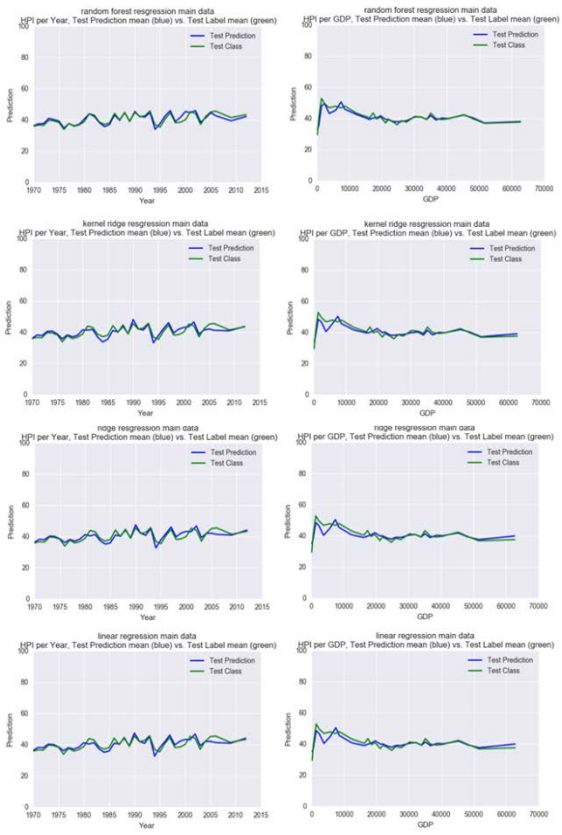
A guide we used for the map-plots generation: http://ramiro.org/notebook/basemap-choropleth/

APPENDIX

RESULTS GRAPHS FROM THE MODEL EVALUATION PHASE, CELLS 8.1-8.3 FOR EACH MODEL:

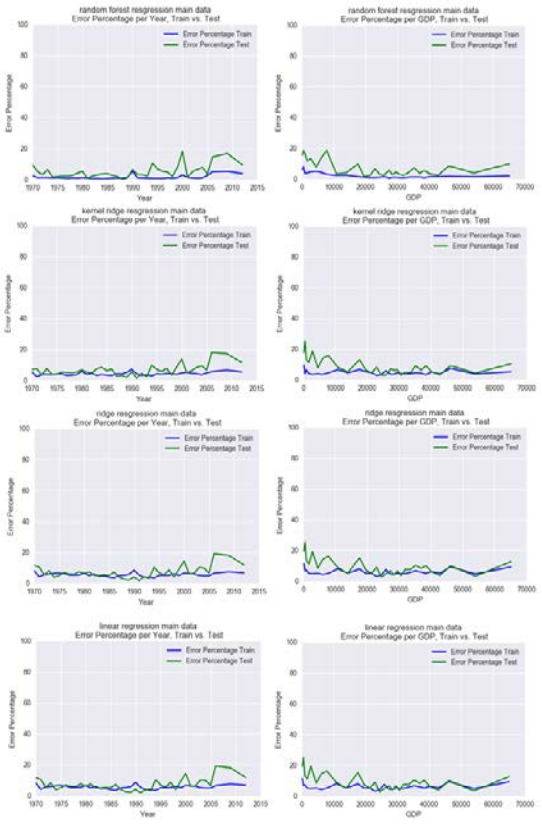
HPI per Year, Test prediction mean (blue) vs. Test label mean (green) for each model as seen in the notebook

HPI per GDP, Test prediction mean (blue) vs. Test label mean (green) for each model as seen in the notebook

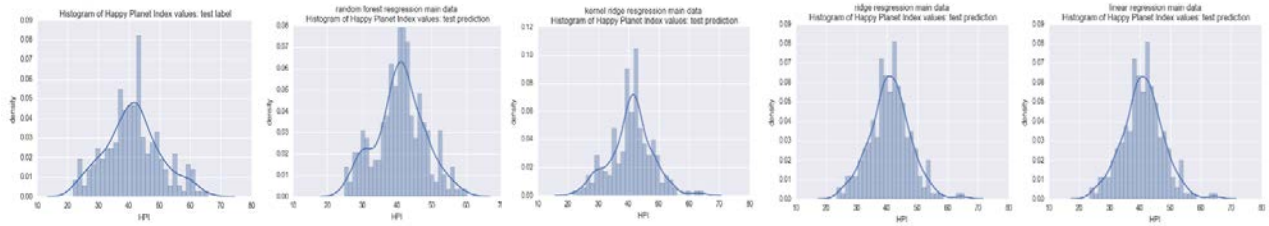


Error percentage per Year, Train (blue) vs. Test (green) for each model as seen in the notebook

Error percentage per GDP, Train (blue) vs. Test (green) for each model as seen in the notebook



Histogram of HPI for each model as seen in the notebook



NUMERIC RESULTS COMPARISON FOR ALL MODELS, CELL 8.4:

	Parameter	Linear Regression	Ridge Resgression	Kernel Ridge Resgression	Random Forest Resgression
0	R^2 for Train data	0.869075	0.869071	0.907824	0.967862
1	R^2 for Test data	0.615158	0.615270	0.645739	0.718918
2	Mean HPI for Train data	41.055932	41.055932	41.055932	41.055932
3	Mean prediction for Train data	41.055932	41.055932	41.057410	41.051285
4	Mean HPI for Test data	41.364372	41.364372	41.364372	41.364372
5	Mean prediction for Test data	41.027071	41.027066	40.927723	41.001483
6	Error Percentage for Train data	6.090462	6.089859	4.927728	2.444418
7	Error Percentage for Test data	9.884222	9.883660	9.372737	7.627692

PREDICTION OF RANDOM FOREST ON THE TEST DATASET COMPARED TO LABEL, CELL 9:

TYPES OF DATASETS' PREDICTION COMPARISON, CELL 10:

	parameter	main data	no countries	no years
0	R^2 for Train data	0.967862	-0.044004	0.961483
1	R^2 for Test data	0.718918	-0.103774	0.448158
2	Mean HPI for Train data	41.055932	40.815178	41.034000
3	Mean prediction for Train data	41.051285	41.513766	41.014139
4	Mean HPI for Test data	41.364372	41.364372	41.364372
5	Mean prediction for Test data	41.001483	40.946224	43.033926
6	Error Percentage for Train data	2.444418	14.163411	2.714507
7	Error Percentage for Test data	7.627692	17.750404	10.303969