IN3062: Introduction to Artificial Intelligence

Summer Coursework

Wojciech Kilian

Domain and dataset

From a broad range of interesting datasets, and domains resulting from them, I decided to choose Life Expectancy factors (WHO) dataset. As I learned the module I realized that I will feel more confident while working on numerical or text values from statistics datasets, rather than image or voice recognition. This particular dataset seems interesting to me because of matching data type, real-life data and number of questions that can quickly arise when analyzing it, and therfore coming with correlations to analyze and predict. Dataset contains 22 features and 2938 rows of samples.

Questions and Tasks

The questions I asked myself and decided to try answering, when I started analyzing this dataset, were:

Which of these features have the biggest impact on expected life length and how are they correlated?

Given this data, how well can models predict it?

Are all the features having the same impact on expected life length for all the countries?

Tasks that were results of these questions were: Calculating Pearson's correlation coefficient for all valuable features, Creation of 2 regression models that will predict life expectancy, Ploting the graphs for top 3 most corelated or influential features

Data processing steps

- Analyzis of features and their initial dependencies

- Droping features with high dependency to prevent bias

- Transformation of text features into numerical data or droping them if unnecesary

Data preparation

First of all I decided to check and make sure I understand what each feature actually is. I quickly found out that one of them - Income composition - is part of of Human Development Index.

According to United Nations Develpment Programme,

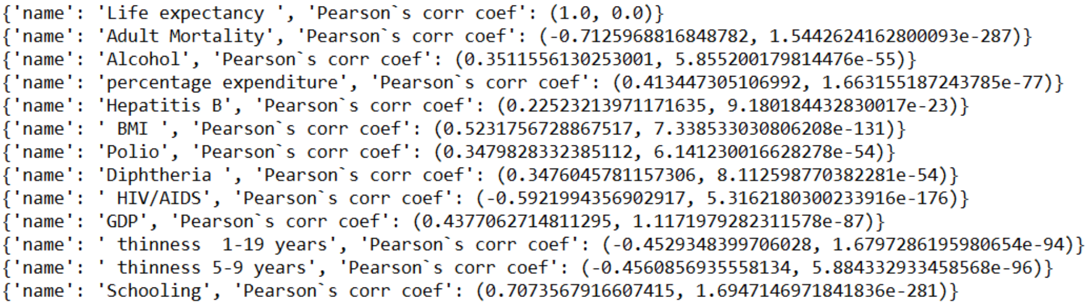
"The Human Development Index (HDI) is a summary measure of average achievement in key dimensions of human development: a long and healthy life, being knowledgeable and have a decent standard of living. The HDI is the geometric mean of normalized indices for each of the three dimensions.

The health dimension is assessed by life expectancy at birth, the education dimension is measured by mean of years of schooling for adults aged 25 years and more and expected years of schooling for children of school entering age. The standard of living dimension is measured by gross national income per capita. "

As it was using 3 other features, all of which we already have data for (life expectancy, Schooling, GDP is partially GNI), I decided to remove it from calculations (comment in code has been left).

Another feature worth mentioning is Adult Mortality. As name implies, it will have strong connection with Life Expectancy, however, it is not a direct cause, but more of a result. Nevertheless, as same countries during various years had different Adult mortality rates I decided to leave it in calculations.

Then I calculated Pearson's correlation coefficient to answer which of features have highest correclation and high probabilities for not being random. I checked the values with filling missing gaps with median and with method of dropping rows. PCC results were slightly closers to extreme's when I used dropping so I decided to use dataset prepared this way in first model - Linear Regression. Then, I decided to leave text type columns out of calculations. I also prepared versions with exclusions of Population data (PPC:-0.02) and Total expenditure (PPC:0.17) as they seem to me too low to bring positive effect.



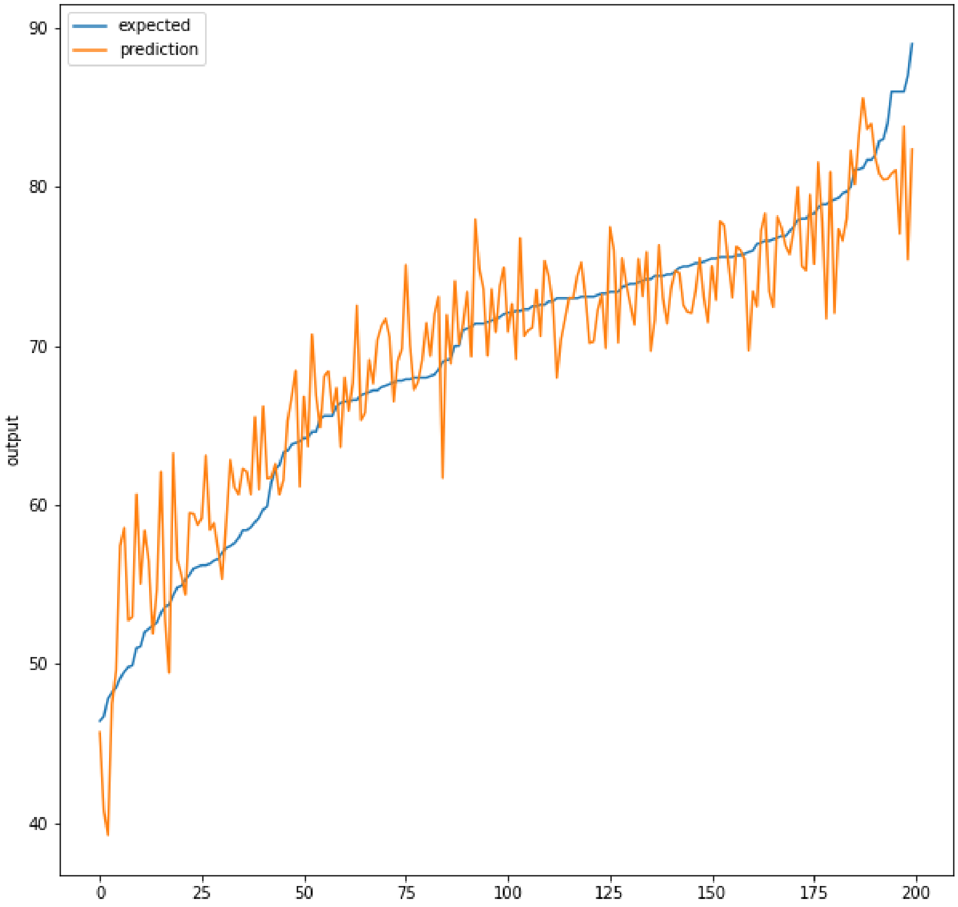
```
{'name': 'Life expectancy ', 'Pearson`s corr coef': (1.0, 0.0)}
{'name': 'Adult Mortality', 'Pearson`s corr coef': (-0.7125968816848782, 1.5442624162800093e-287)}
{'name': 'Alcohol', 'Pearson`s corr coef': (0.3511556130253001, 5.855200179814476e-55)}
{'name': 'percentage expenditure', 'Pearson`s corr coef': (0.413447305106992, 1.663155187243785e-77)}
{'name': 'Hepatitis B', 'Pearson`s corr coef': (0.22523213971171635, 9.180184432830017e-23)}
{'name': ' BMI ', 'Pearson`s corr coef': (0.5231756728867517, 7.338533030806208e-131)}
{'name': 'Polio', 'Pearson`s corr coef': (0.3479828332385112, 6.141230016628278e-54)}
{'name': 'Diphtheria ', 'Pearson`s corr coef': (0.3476045781157306, 8.112598770382281e-54)}
{'name': ' HIV/AIDS', 'Pearson`s corr coef': (-0.5921994356902917, 5.3162180300233916e-176)}
{'name': 'GDP', 'Pearson`s corr coef': (0.4377062714811295, 1.1171979282311578e-87)}
{'name': ' thinness  1-19 years', 'Pearson`s corr coef': (-0.4529348399706028, 1.6797286195980654e-94)}
{'name': ' thinness 5-9 years', 'Pearson`s corr coef': (-0.4560856935558134, 5.884332933458568e-96)}
{'name': 'Schooling', 'Pearson`s corr coef': (0.7073567916607415, 1.6947146971841836e-281)}
```

Dataset split, use and results

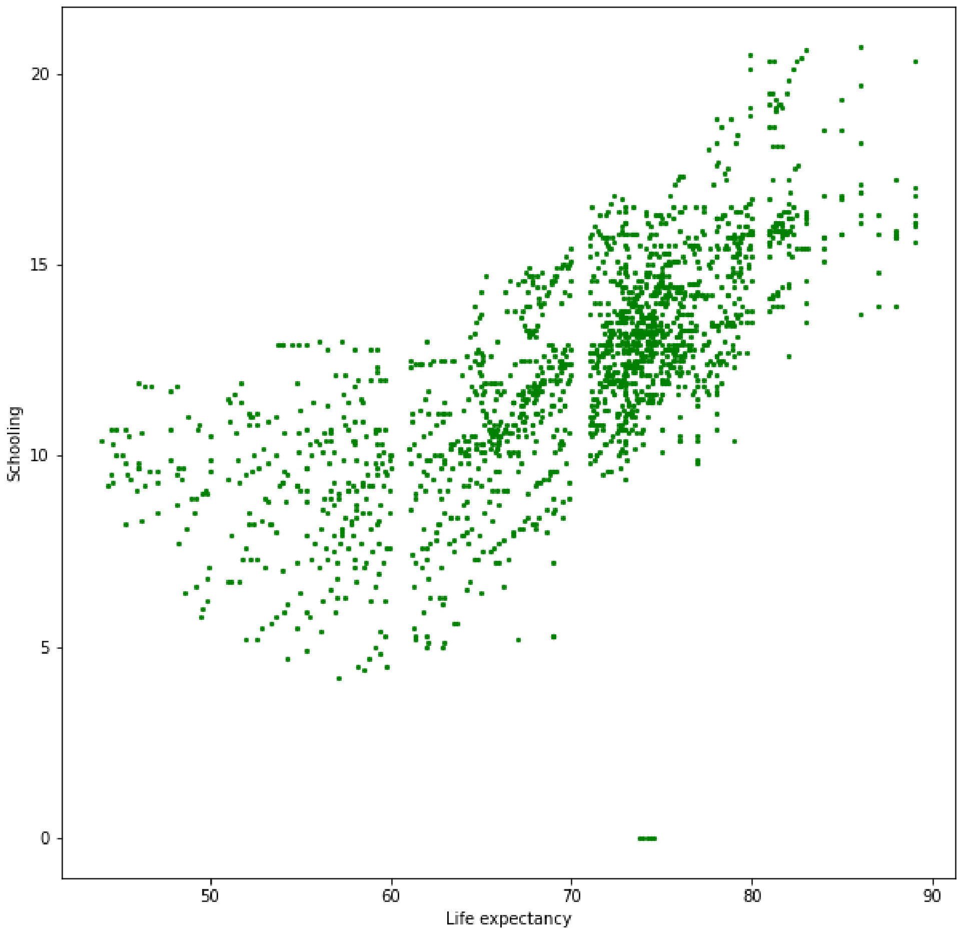
Model 1 - Linear Regression

Due to popular use of train_test_split function from sklearn module I decided to use it both in Linear Regression and Neural Network. Initially I was considering Kfold, but this option was giving me almost instant results to work with and improve model. To establish best model I decided to run through 5000 random seeds and set 2 best: one giving highest R^2 and one giving lowest ratio of RMSE to Mean; According to lecture information percent of of RMSE to Mean below 10% would be satisfying. After fine tuning it I've dropped from 10% ratio to 4.5. Then I started looking for seed with best value in 3 instances: with population feature, without population, and without population+total expenditure. Results were really close to each other in all 3 instances, but using the last one (Population data (PPC:-0.02) and Total expenditure (PPC:0.17)) gave a tighter graph of expected vs actual output prediction. Also, initially I was checking results on data with gaps filled with median and dataset with dropped fields, but the second one was giving better results (Lower RSME, higher R^2) - code to prepare dataframe with filled gaps is commented out but availible at the bottom of the page.

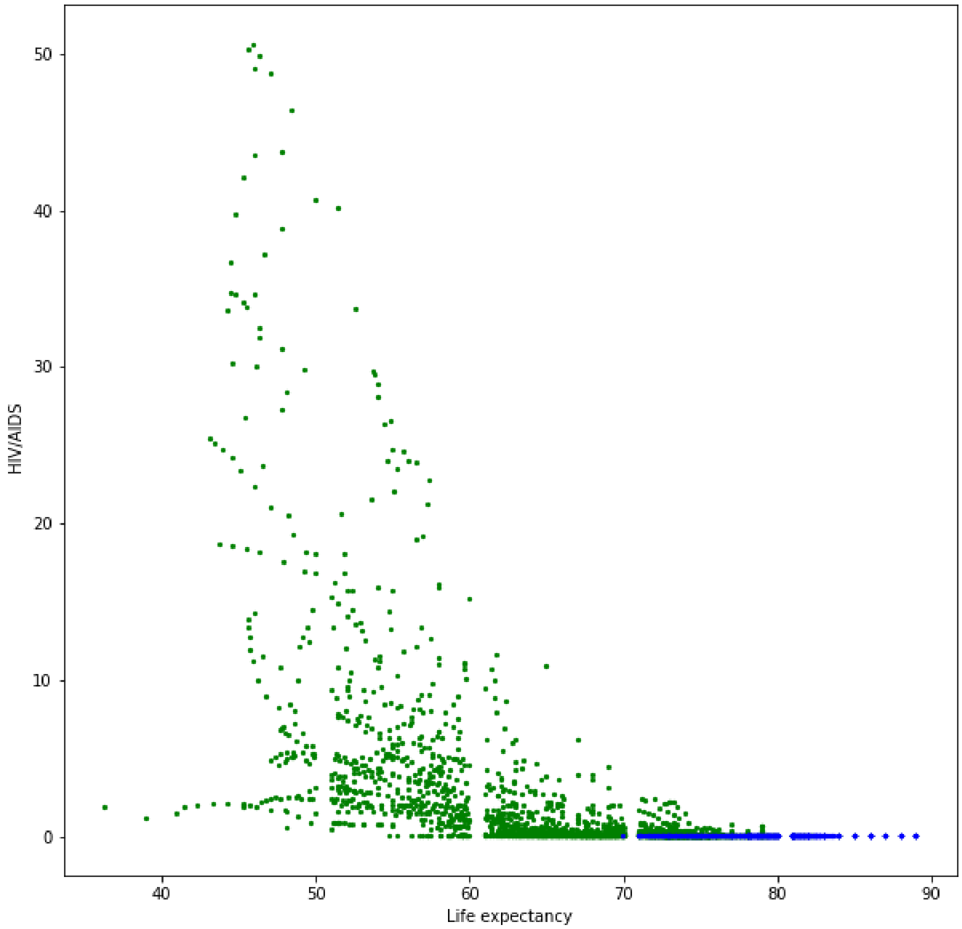
Lines 1-28: Loading of dataset, and discarding collumns, L 30-50: Preparing Dataframe and Pearson's corelation coefficient, L 50-65: Dividing dataset, L 66-78: best scores L 80-125: Uncomment this to see the calculations of best model (by change of randomstate), L126+ Producing the best model and resulting graphs.

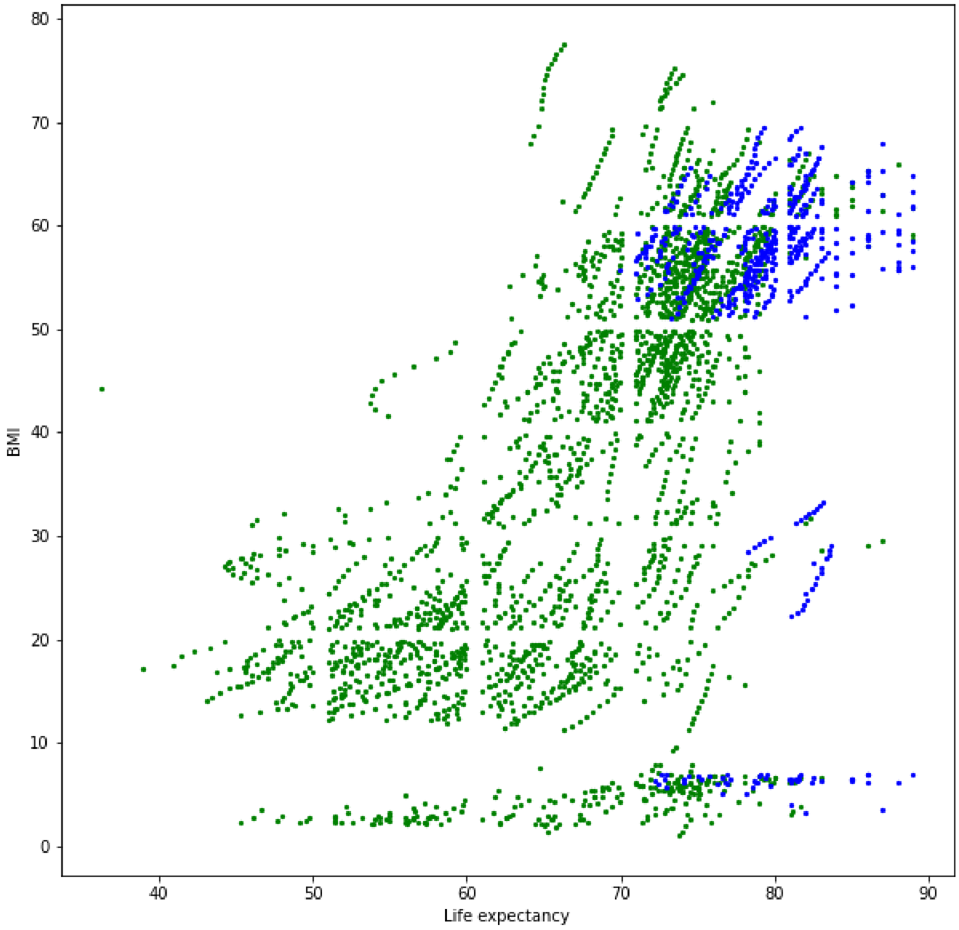


Code will produce 7 graphs, first 3 are representing how well prediction fits the data (and doesn't overfit), depending on number of points. Next 3 are showing Top 3 correlations to life length prediction : (PPC: 0.73) Schooling, (PPC:-0.55) HIV deaths among children 0-4, (PPC:-0.54) average BMI of country population.



(Countries "Developed" in blue on graph 2 and 3)

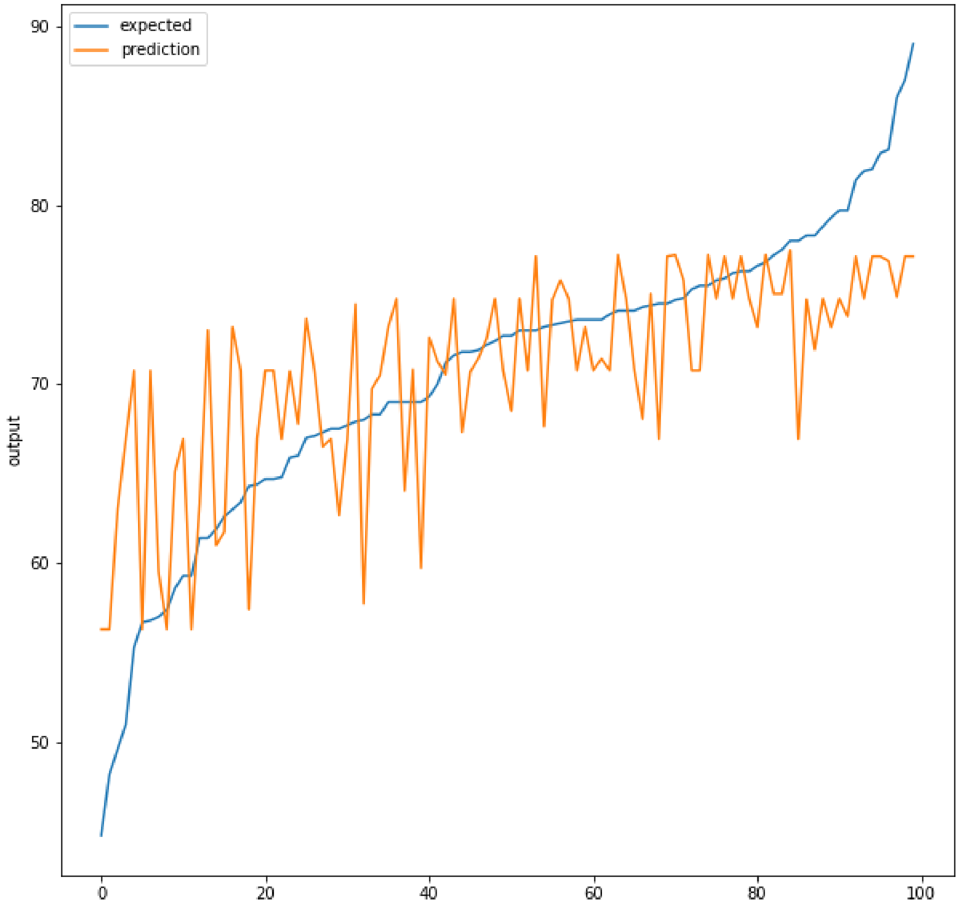

HIV/AIDS
50
40
30
20
10
0
40
50
60
70
80
90
Life expectancy



Last graph shows (PPC:-0.7) corelation of Adult mortality - more in: Data Preparation paragraph.

Model 2 - Neural Network regression

Here I didn't change random seed. I decided that tuning the model myself, step by step, and using Dropout to randomly turn off neurons will be enough to ensure different results. I later tried using Kfold=5, but the results between folds were not that different and my RSME was still high. I double checked if all data is numerical and used helping function to convert status of countries to 0-1 values. I started bulding model with few neurons in 1 layer and by growing its number in layer, I decided results are not getting much better after number of neurons was more than 2 times number of features (~20). Then I tweaked number of epochs and quickly got to conclusion my results are not getting better after 80 epochs, so I set them to 100 and decided to test other activation functions, as well as started thinking about implementation of early stopping. After testing many options, I added another layer and tried combinations of different activation functions. Finally from initial 100 loss i dropped to below 30 with use of: 3 layers + 1 dropout layer for regularisation, combination of sigmoid and ReLU functions, kernel regularizer to prevent overfitting, early stopping monitoring, usually in half of epochs or below. I'm surprised I wasn't able to get lower results, even after adding momentum, as for a long time I was sure I reached local minima.



Reflection

Project:

At the beggining of analyzing the data I had my intuitional picks on which features from this dataset will have biggest impact on life expectancy and which won't. I am certainly surprised that biggest coefficient was with schooling, as I didn't expect education to have such a big impact on life, from this physical side. Second most interesting thing was influence of HIV on overall correlation; as we see on second graph, all countries with high HIV cases number are behind all the developed countries which have no HIV cases on that scale. And what left for me was appearance of BMI index on 3rd place. I was confident it will be top corelation to health and life expectancy, as obesity is one of the biggest silent killers. Besides that I expected higher influence of GDP in positive corelation and Alcohol with much more negative.

From technical side I was expecting myself to do more on Model 2 and less on Model 1, but it turned out that I was able to prepare, fit the data more accurately and achieve better results

I am certainly happy that my codes worked and I could answer questions I gave to myself. I am aware that there is a lot of place for improvement, that different, more elegant techniques could have been applied to achive the same or better results. As I'm repeating this module, I am now quite confident about AI theory topics, maybe less confident on practical side, but I assume it come from lack of practice in Python and smaller time spent to learn details of the module.

References

I created models 1 and 2 mostly by using knowledge from jupyter tutorials: Accordingly 3 and 6. To be completely sure I don't get bugs, that ocassionally appeared, some lines of code are directly copied from tutorial and therefore I will cite them.

Model 1 includes: basic statistic printing from tutorial 1 (line 40); defining feature data the same way as in regression tutorial 3.1(line 58); regression comparsion tutorial 3.1; chart_regression function for drawing graph(line 149-158);

Model 2 includes: 2 helping functions: to_xy(to convert data) and encode_text_index(to encode status of countries to 0-1) (line 28-44); step-by-step creation of Neural network according to https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/ and tutorials 6,7,8(line 72-80); reuse of chart_regression function for drawing graph(line 101-113).

Other outer references

Dataset: https://www.kaggle.com/kumarajarshi/life-expectancy-who

Omiting HDI (preparation): http://hdr.undp.org/en/content/human-development-index-hdi

Pandas API reference: https://pandas.pydata.org/pandas-docs/stable/reference/index.html

Regression tutorial(comparsion/debugging):
https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/