Introduction to Data Science - Project Report

Death statistics

Topi Laanti, Olga Viholainen ja Selina Lehtoranta

Overview

Initially, we set ourselves a somewhat vague goal. We found a great dataset that had massive amount of death statistics data, and we thought it would be cool to use it and some other data and look into if it's possible to apply the techniques we've learned during the course find out something useful!

First we thought about choosing a few causes of death, such as suicide-related causes or diseases, and multiple economic factors look into if it's possible to predict how well we can predict the ratios of these health-factors based on the economic ones. But we quickly found out that it would be best to focus on only one health cause.

Wrangling like a proper Data-Cowboy and taking a look into your cattle

We downloaded the data and to surprise of none - it was massive. WHO:s data turned out to have over 3 million rows of data which meant that we soon ran into one of the first problems I assume many fledgling data scientists ran into: how do we handle this data without it taking hours? All of us are beginners in Python and data wrangling in general, so needless to say we were puzzled. Our initial attempts at combining the data resulted to our computers seemingly freezing, and we thought about letting one of our computers to run overnight and afterwards just saving resulting data to a new file. After many hours of googling, we found out there was a pandas function called df.concat, which did the thing we wanted very efficiently. *Oops.*

As we started to look into the data, we found out that it was extremely rough and varying. The dataframe included data from countries that haven't existed for many years. Similar death causes were coded differently in different countries, and some countries had barely any data at all. So the obvious solution was to just choose one field of death causes - something that would be telling of a nation's state, so we ended up with choosing suicide related death causes.

Suicides and depression go hand-in-hand and they are one of the plagues of welfare states of world's countries. It's obvious that undeveloped countries have more trouble with diseases such as malaria or HIV, and just having a poor economy reflects the higher rate of those, so we didn't find researching that as interesting as figuring out what differences rich countries that have different rates of suicide seemingly have.

Making sense of more than our feelings

The more and more we got familiar with the data, the more we started seeing issues in it. Countries were represented with awkward codes that made it hard to figure out what are we looking at, so the next step was naturally to modify the country names to their proper names. We found a PDF from WHO:s website that contained all the codings, and we found out that basically comparing every row of a dataframe with 3 million rows to a list of keys took a huge amount of time! It took a fair amount of time to figure out how to do this efficiently, but before that we decided it's probably best to take subset of the data, which meant that it would be easier to continue with our project.

At this point we realized it's probably best to get the other dataset our project needs, the dataset that had economical and well-being factors. After furious googling, we found out a great dataset that had multiple different variables that all were related to economy and wellbeing from many countries and years. The only issue was that it consisted of only european countries, which in reality wasn't an issue at all and we were quite happy to only look into european countries!
We dropped all rows that were not EU countries by creating a list of the desired countries and then comparing it to the dataframe.

EU is infamous of its bureaucracy and we found out that they have fantastic datasets of all sorts of different factors. This time the data wrangling was easier, since we could apply the same techniques we learned before to transform it into a nice form. This included changing country codes to country names, imputing missing data, converting the data to the right variable type, and creating a new dataframe that had a row per every cell of the old dataframe. The same method could be used to every EU dataset, after which you combined all of them into a one big dataframe, that could be used with the suicide dataset.

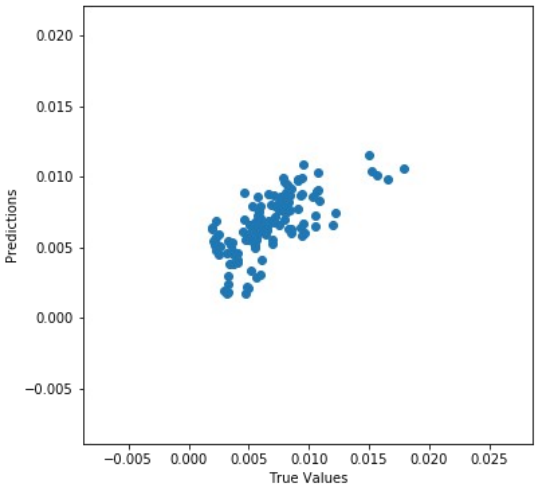
Creating a model

Our data:

	country	year	poor retirees	employment	debt	lost children	economy_equality	education	suicides
0	Austria	2011	15.5	74.2	82.4	8.5	4.1	82.4	15.54
1	Austria	2012	14.4	74.4	81.9	8.2	4.2	82.9	15.20
2	Austria	2013	14.6	74.6	81.3	8.6	4.1	83.0	15.35
3	Austria	2014	14.2	74.2	84.0	9.3	4.1	83.9	15.26
4	Austria	2015	12.9	74.3	84.6	8.7	4.0	84.6	14.48
5	Belgium	2011	17.3	67.3	102.6	13.8	3.9	71.3	19.25
6	Belgium	2012	16.7	67.2	104.3	14.4	4.0	71.6	18.76

Poor retirees: The at-risk-of-poverty rate for pensioners. The target of the rate is zero.
Employment rate: employment rate in age group 20-64.
Debt: General Government Gross Debt to GDP.
Lost children: The youth-down-and-out indicator measures the share of the population aged 15 to 29 who is not employed and not involved in education or training. The target of the indicator is zero. **Economy equality**: The Income quintile share ratio measures the ratio of total income received by the 20 % of the population with the highest income to that received by the 20 % of the population with the lowest income. Big values mean strong inequality. **Education**: share of at least upper secondary educational attainment.
Suicides: the amount of suicides per 100 000 inhabitants.
Variables: poor retirees, employment, debt, lost children, economy equality and education were our explanatory variables and suicide was our target variable.

Our data consisted of only 140 data points, because we had difficulties to find good data. We wanted anyway to create a model, estimate the strength of the relationships between our explanatory variables and target variable and predict the suicide rate, but we knew that our results would hardly be impressive. First we normalized our data, dropped columns *year* and *country.* Then we splitted the data randomly to training data (70%) and test data (30%). We fitted the linear model that used method:
ordinary least squares. We did this part five times and counted means, because we wanted to avoid overfitting.



Results:

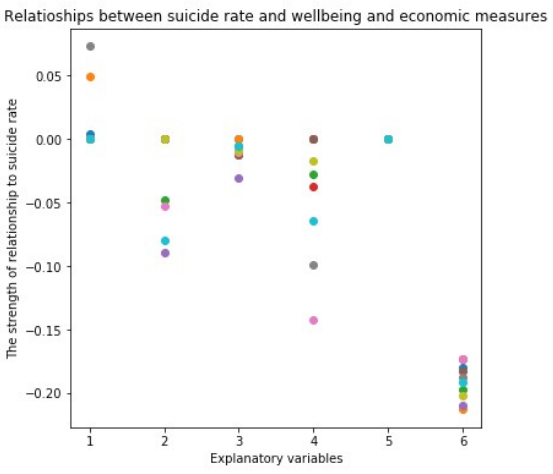
```
Average test score:  0.45863610539128297
Average training score:  0.49554375241482373

              Average coefficient:
poor retirees      0.1554
unemployment      -0.8619
debt      0.0007
lost children      -1.0459
economy equality      1.1678
education      -0.3130
```

Our test and training scores were moderate, maybe better than we expected. We expected that variables: *poor retirees, debt, lost children* and *economy equality* would have positive coefficients and variables *employment* and *education* would have negative coefficients. It was very weird that the variable *lost children* had a negative coefficient. We thought could this be a result from multicollinearity or did this happen because we didn't have a lot of data?

The coefficients of the variables *poor retirees* and *debt* were quite near zero. We had thought that *debt* would hardly have a great influence on the suicide rate but the coefficient of *poor retirees* was a surprise for us. We decided to use Lasso regression that we better avoid overfitting and get rid of multicollinearity.

Lasso: we splitted data randomly to training and test sets and fitted the model many times. Results with alpha = 0.00003:



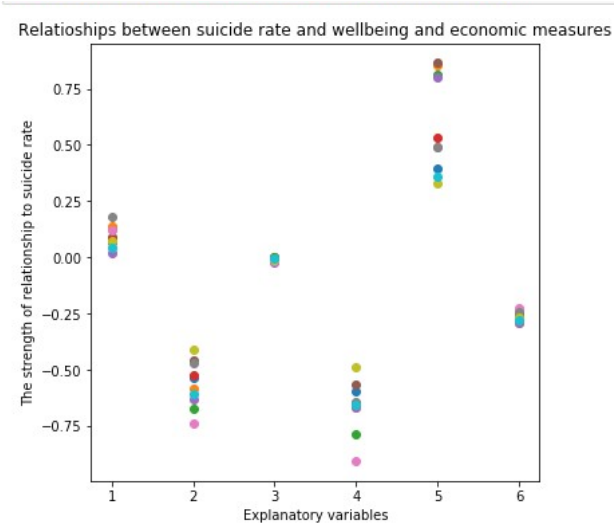

```
Average training score: 0.3436
Average test score: 0.2834
```

1 = poor retirees
2 = employment rate
3 = debt
4 = lost children
5 = economy equality
6 = education

Here you can see the coefficients of every variable with alpha = 0.00003. The variable 6 *education* seemed very important, because it's coefficient was never zero. Each lasso model had set 4-3 coefficients to zero. With

alpha = 0.00003 the average training and test scores were quite low, so we had to make the alpha smaller.

Results with alpha = 0.00001



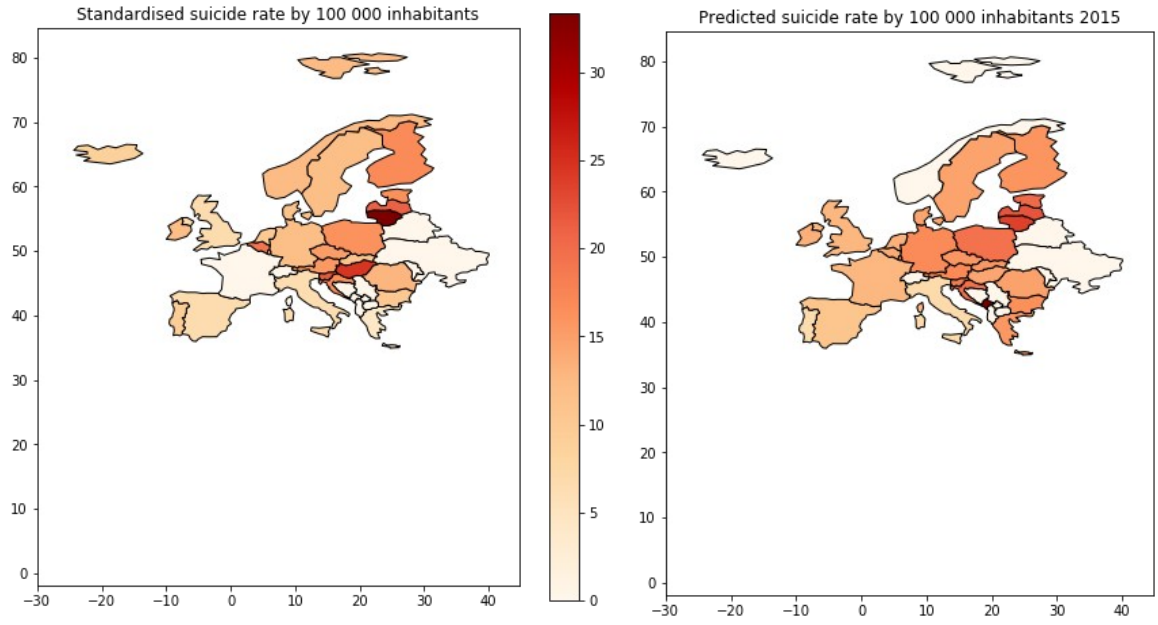

```
Average training score: 0.4749
Average test score: 0.3842
```

1 = poor retirees
2 = employment rate
3 = debt
4 = lost children
5 = economy equality
6 = education

Here you can see the coefficients of every variable with alpha = 0.00001. Every model had set 0-2 coefficients to zero. With alpha = 0.00001 the average training and test scores were better. More than a half of the these lasso models had set the coefficient of the variable 3 *debt* to zero, so we decided to remove this variable.

Final model

Our final model had five explanatory variables: poor retirees, employment rate, lost children, economy equality and education. We tried also to predict the suicide rate with our model. Our result wasn't very good… The score of our model was about 0.401



Appendix

We collected data from WHO and Eurostat and combined them into one dataset using only variables we chose to study. Data was easily available, but it was hard to wrangle it into a proper form considering our experience in python. It took a long time to understand the sentimentals of pythons tools, but after hard work we started to get results and be comfortable with python. Our group work consisted mostly of us three gathering together and brainstorming and then executing our visions. We used mostly pandas, numpy and matplotlib packages and in addition geopandas to create beautiful maps and sklearn to create linear model and lasso model.

Bibliography

Eurostat, 28.10.2018, *https://ec.europa.eu/eurostat/data/database*

WHO, 28.10.2018, *http://www.who.int/healthinfo/mortality_data/en/* and *https://www.who.int/mental_health/suicide-prevention/en/*