Predicting Life Expectancy

Motivation: when you wish to save for your retirement provision, insurance companies are interested in predicting how much will you live in order to estimate how much you should pay to them. For example, if you end up living way longer than it was expected from the insurance, the insurer will have a € loss.

For this project I will have a set of predictors (e.g. income, spending on healthcare, etc) to estimate life expectancy (the target). As we have a label (life expectancy) and this label is within a range, we are dealing with typical scenario of Supervised Machine Learning – (Linear) Regression.

1) Exploring and cleaning the data

- First dataset (source: Kaggle/WHO 2000-2015)
    - I start by seeing the **shape** (~3K entries, where each entry is a country on a given year) and the **type** of data (only numerical) and **proportion of missing values** per column.
    - I choose to keep only the columns that give me information per capita (e.g. Years of Schooling, Income, etc) rather than rates (Adult Mortality, Infant Deaths, etc), as this is the data that will be relevant for the client (aka insurance company) algorithm.
- Second dataset(s) (source: gapminder.org)
    - As I end up with very little columns due to my 'per capita' filter, I decide to search for more predictors (C02 emissions, Children per Woman, Sugar Consumption per Day, etc).
    - Each of these predictors are a unique data-frame, where columns are years and each row is a country. In order to merge with my first dataset, I:
        - Melt the data in order to have just one column (e.g. values for Kcal Consumed per Day) and each row is a country in a given year.
        - Dealt with the different naming for countries (e.g. USA/America, Republic of Moldova/Moldova, etc).
        - Make sure I was dealing with same range of years (some data I only found from 2005 till 2015 so I had to work with that range for all the predictors – inner joint).
- Both sources (Kaggle and gapminder.org) had values for life expectancy. Before merging the two datasets, I made sure that these values were similar/in agreement (Goodness of Fit: I want to prove that both distributions fit, and there are [p-value]% chances of me being wrong).
- Finally, I merge the datasets and I check for missing values: some I drop (low percentage), some I filled (example: years of schooling for North Korea were missing in the original dataset and I searched what was this value for, say, 2015 and I filled missing values with it for North Korea).
- I run a correlation heatmap and inspect correlations (could be that some columns were to be dropped due to high collinearity/they will correlate with the target the same way).

2) Data visualization

For this project I didn't do much of EDA (Exploratory Data Analysis) simply because I didn't need it. The datasets were pretty simple and straightforward. Would be interesting to plot the evolution of some predictors over time and compare between countries. But that is something that was already done by Hans Rosling on his website gapminder.org, and also was not what I was looking into in my project.

3) Machine Learning

Basically, here I split the data into two sets: one for training the model (80%) and other to test it (20%). As I am dealing with a continuous target (life expectancy is continuous data), I went to explore the regression algorithms (for Supervised Learning) we learned in class: Linear Regression, Decision Tree and K-nearest Neighbour. The Decision Tree model gave me the best accuracy (91.33%) so I used this model to my final step.

4) Final step

Finally, I built an online app (a very rudimentary app as my skills of web development are not the strongest) with the help of www.pythonanywhere.com, where a user (e.g. an insurer) can input her/his predictors (years of schooling, grams of sugar consumed per day, etc) and have her/his life expectancy predicted. You can find the app here: http://filipamiralopes.pythonanywhere.com

Filipa Lopes
Project 6
2nd October 2019
#data-squad-21
Ironhack Lisbon