Visualization of Massive Data

Project

by Esteban Lewis

Contents

1. Project introduction

The goal of the project is to use the Python language and related libraries to create a visual representation of a relatively large amount of data, and to learn a predictive model of one attribute based on others.

For this project, I have chosen a dataset of diagnostic measures related to diabetes in Indian women, and have decided to use this dataset to create a predictive model that would allow us to predict whether someone is likely to have diabetes or not.

2. The dataset

The dataset contains information about Indian women who have and don't have diabetes. The attributes (columns) are:

- Pregnancies: how many times has the woman been pregnant
- Glucose: plasma glucose concentration in an oral glucose tolerance test
- Blood pressure
- Skin thickness: triceps skin fold thickness
- Insulin: 2-hour serum insulin
- B.M.I.: body mass index
- Diabetes Pedigree Function (D.P.F.): a function which scores likelihood of diabetes based on family history
- Outcome: whether or not the woman has diabetes

This dataset can be found here:

https://www.kaggle.com/uciml/pima-indians-diabetes-database

3. Reduction of data

The Python script I created gives the user the possibility to reduce the data by randomly selecting a given fraction of it. For instance, if the user wants to analyze only 50% of the data, then the script will only select 50% of entries randomly.

Additionally, the user can also choose to apply a lower limit and an upper limit to a given attribute of the dataset. For instance, the user may choose to analyze only patients between 40 and 50 years old.

4. Visualization methods

I have decided in a first place to create a scatter plot matrix using the Python library matplotlib, in which all healthy patients are represented in green, and patients with diabetes in red. This allows us to clearly see how some attributes are connected to likelihood of diabetes, and also how two attributes may be connected to each other.

The second visualization method offered to the user is a parallel coordinates plot, once again created by matplotlib. The user can choose up to 3 attributes (not including "outcome"), and the script will create a parallel coordinates plot where we can see how those three attributes are connected to each other and to the likelihood of diabetes. In the parallel coordinates plot, the user may choose to scale the data in order to avoid some columns being much smaller than others (for instance, it would be difficult to visually compare age and age and DPF, considering that age goes up to 80 and PDF always stays below 1).

If you wish to create a parallel coordinates plot, it is recommended that you select only a part of the dataset (see 3. Reduction of data).

5. Visual results

On the created plots, we can see that most attributes are somewhat related to the risk of diabetes. Some, such as the glucose test, appear to be highly connected to the risk of diabetes. Others, such as blood pressure, seem to have very little to no link with diabetes. However, we can see on both visualization methods that no attribute gives a 100% chance of diabetes. We can only make an estimation of the risk, but not give a sure prediction.

Looking at the scatter matrix or the parallel coordinates plot, we can also notice correlations between other attributes of the dataset. High BMI seems to cause thicker skin at the triceps, which, together, are linked to a higher risk of diabetes. There might be relations between other attributes, such as glucose and insulin, but these are less certain.

6. Prediction model

For the prediction model, I have chosen to create a decision tree using the library sklearn. The obvious risk here is overfitting: since people are all different, we cannot draw conclusions based on a handful of patients, and thus we have to limit the size of the tree. For this reason, I have limited the number of samples per leaf to 40 patients minimum. To further avoid overfitting and meaningless tests that make very little difference in the tree's prediction, I have also given the tree a minimum impurity decrease.

By reducing the size of the tree in this way, I have stripped it to only meaningful tests, thus creating a prediction model for diabetes that is more likely to be realistic. The user may visualize the tree by running the script. The Python library "graphviz" is used to save the tree into a PDF file.

7. The script

You may download the Python script here:

https://github.com/lewisesteban/visualization

Required libraries:

- numpy
- matplotlib
- csv
- sklearn
- graphviz

To run the script, you also need to install the graphviz executable and add its path to the PATH variable of your system.

The script requires the data file to be located at the execution path. That data file must be called "data.csv.". To run the script, execute it with no arguments first, and you will be given the list of expected arguments. Run it again with the arguments you want, and follow the instructions given to you.