Introduction to Data Science Long Assignment 2024/2025

Pedro G. Ferreira

October 6 (version 9.10.2024)

Predicting Health Insurance

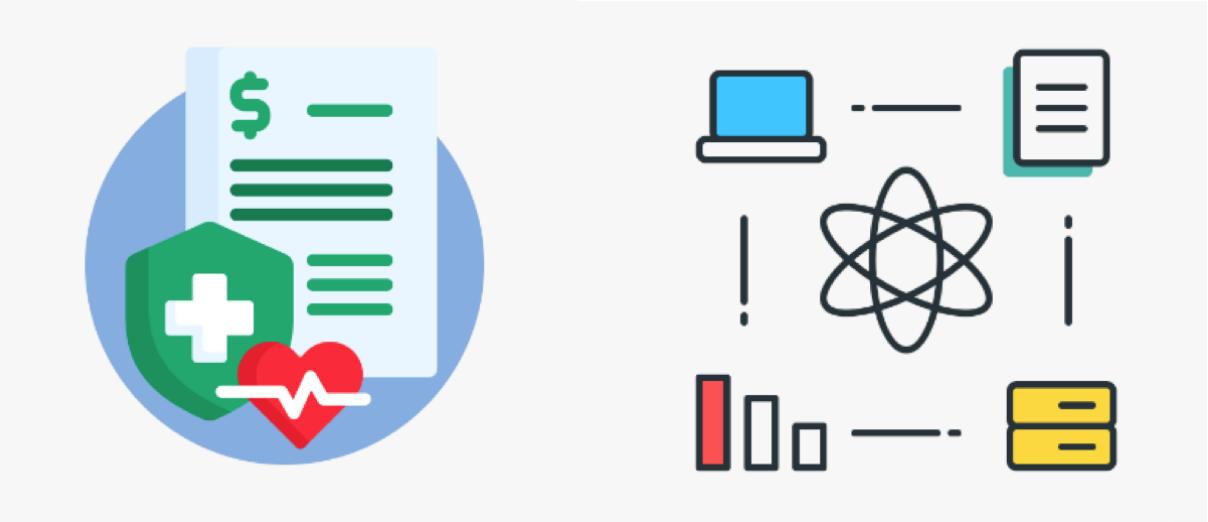

Figure 1: Applying Data Science to Predict Health Insurance

Suppose you are working as a data scientist for a company and you are given a project to build a model to predict which of your customers don't have health insurance. You've collected a dataset of customers whose health insurance status you know. You've also identified some customer properties that you believe help predict the probability of insurance coverage: age, employment status, income, information about residence and vehicles, and so on.

For this assignment you will **build and test** a predictive model that given a dataset called *customer.csv* will predict if a customer has or not a health insurance. This is a database from US customers, therefore the health system is different from most of the systems in European countries.

Remember that no dataset is perfect! Before you start modelling, get to know your data very well. Take care of inconsistent or incomplete data. If you don't take the time to examine the data before you start to model, you may find yourself redoing your work repeatedly as you discover bad data fields or variables that need to be transformed before modeling. In the worst case, you'll build a model that returns incorrect predictions—and you won't be sure why.

Use summary statistics and visualizations to spot problems and describe and explore your data.

Dataset

You will receive four files from this dataset:

- customer.csv
- customer_test_masked.csv
- customer_datadictionary.txt
- sample_submission.csv

File customer.csv to explore the data and train the model. The dataset *customer_test_masked.csv* should be set aside for the final test of the model. You should use this dataset only in the very last step of the project. Once you are confident with your model, you can evaluate it on the test dataset. It contains 804 entries, where the field *health_ins* was masked (set to null). With this data you will need to do you predictions. Then export those predictions in a format similar to the file *sample_submission.csv*. Next, go to the Kaggle link below, register and create a team to submit your predictions. You will see the results in the leaderboard.

The dictionary with additional information on the dataset is provided by: *customer_datadictionary.txt*

Goal

The main goal of the project is to create a machine learning model to predict if a customer has or not a health insurance.

Data Exploration

Before the development of the model itself try to develop analysis to answer some of the following questions, regarding the profiles of the customers:

- Are they young, middle-aged, or seniors?
- How affluent are they?
- Where do they live?
- Do you see differences in the income related to age, gender or the marital situation?
- Are the characteristics of the housing situation interrelated (e.g. is the number of rooms related to the housing type or gas consumption?)

Check if there are relationships between two variables, including the target variable (e.g.):

- What is the relation between age and income (suggestion: use scatter plot)?
- Visualize the probability of health insurance by age (suggestion: use binary scatter plots)?
- Visualize the marital status according to the housing type (suggestion: use side-by-side bar charts)?
- How the health insurance distributes according to the marital status (suggestion: use stacked barplots with colored histograms per category)?
- Explore correlations between all variables.

Note that the above points are just suggestions for the analysis and you can explore other aspects of the data that you may think are relevant for this problem. Do not limit yourself to these ideas.

Model training

You will need to create the train, validation and test datasets:

With *customer.csv* do:

1. shuffle the dataset entries.
2. create a first division of the data into train (Tr) and validation (Vs) using a proportion of 80/20 or 70/30.
3. Use the training data for calibration of your models (e.g. using Cross validation). You can use sampling, cross validation and other approaches. Use smaller samples to test ideas and reduce computation time. Use larger samples to obtain more reliable estimates of model performance.
4. The Vs dataset should be used to keep evaluating your models.

Evaluate the stratification of the data for the two classes. While this may not be strictly necessary, it is one possible approach to use. I do recommend starting with a predictive model without stratification.

With *customer_test_masked.csv* do:

5. Use the Ts dataset for the final evaluation of your model.
6. Save the model predictions for this dataset in the format of *sample_submission.csv*.
7. Go to Kaggle and upload your predictions and check your position in the leaderboard.

Note that the datasets will need proper pre-processing. This processing should be consistent for both the *customer.csv* and the *customer_test_masked.csv*.

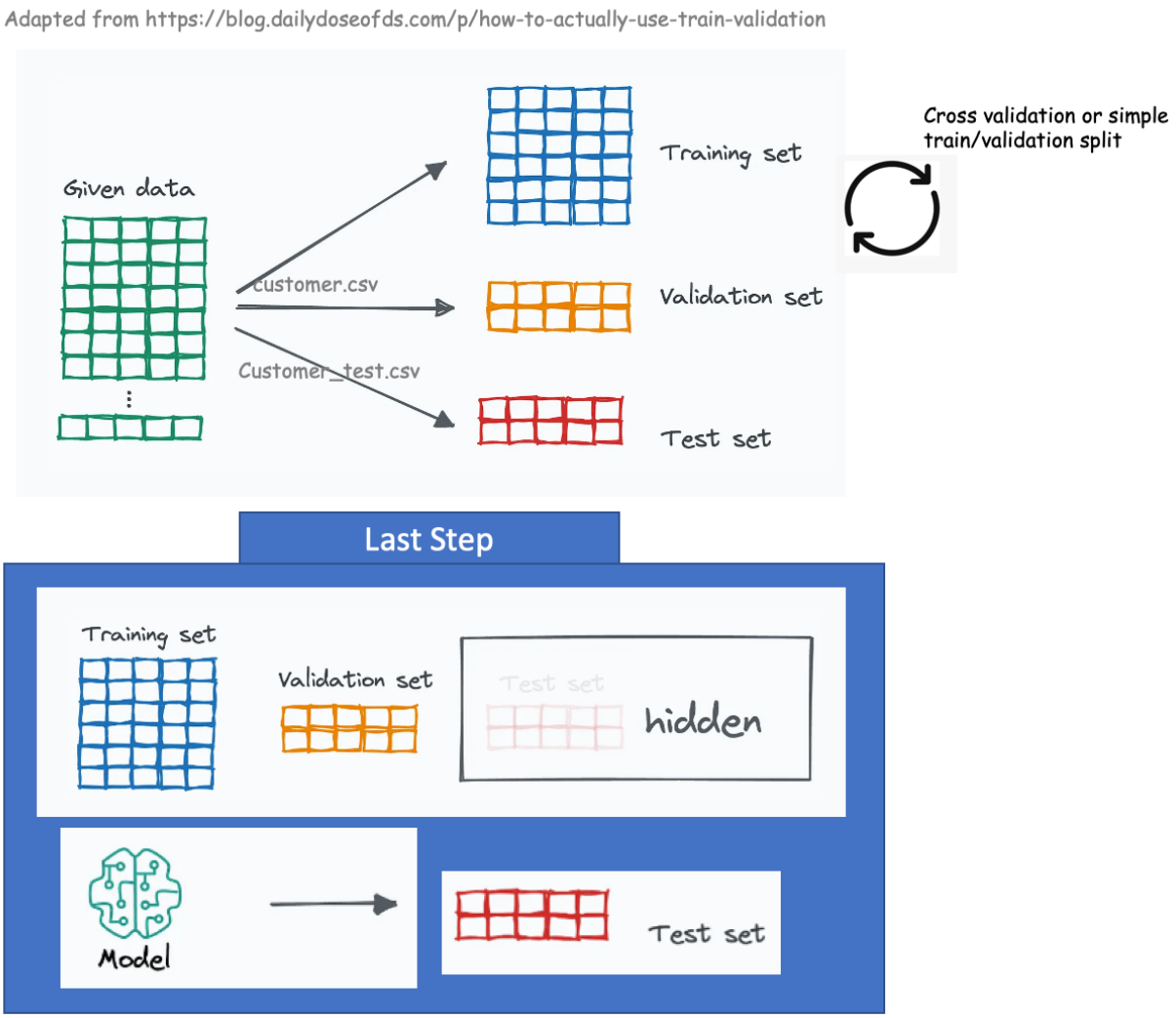

Figure 2: Approach to train and test the model

Check more on this procedure here:

(https://blog.dailydoseofds.com/p/how-to-actually-use-train-validation)

Model Evaluation

Use the different metrics to evaluate your models on the different datasets. Create confusion matrices and other visualizations.

Discuss if the model will be able to properly generalize or presents overfitting evidence.

Guidelines

This data science problem should be approached by following a CRISP-DM methodology (http://jbusse.de/2019_ws_dsci/crisp-dm_phases-tasks-outputs.html) and the data analysis pipelines described in classes.

You have to first understand the business problem, propose success criteria and see how it can be translated into a machine learning problem. Then you look at the characteristics of the data and you perform the required explorations, visualizations and transformations. Next step is to identify insights, develop predictive models and to evaluate them in order to validate if they are helpful in the business problems. During the whole process take notes, always identify the questions you want to answer and think before you act: "why is this plot or this transformation useful". You can perform some operations just for the sake of training but you should be aware of that.

The result is a **report** in the form of a **notebook** with clear explanatory text and code that works showing results. The report should be clear, as concise as possible and it should be easy to read and to follow. You will be telling the story of your approach to this problem, so it should have a good narrative flow. Always explain what you are doing, why you are doing it, what are the results and what do you take from those results.

Suggested structure

A report containing:

1. Business understanding
    - Give your view of the business problem following the CRISP-DM list of outputs when adequate.
2. Data Understanding
    - Looking at the raw data, describe variables according to their types: interval-scaled, binary, nominal, ordinal, ratio-scaled. Be aware that there are specific methods suitable to each type of variable.
    - Perform a preliminary analysis (summaries, spread measures, histograms, boxplots, density plots). These are interesting to be applied to the raw data to "uncover' ' inconsistencies, outliers, duplicates etc.
    - Perform bivariate analysis (correlations, regression)
    - Provide any insights about the data and the problem that you may have found.
3. Data Preparation
    - List of main changes that can need to be performed to the raw data, including feature selection.
    - Describe the potentially useful ones and their results in terms of data.

4. Modeling: consider the balanced and the non-balanced versions of the dataset as 2 separate problems. First work with the balanced data and then with the non-balanced data. Try each of the methods below, select hyper parameters using default values and empirical analysis. Separate a test set and use cross-validation on the rest of the examples. Visualize models when possible, visualize results, produce aggregating tables with good insightful summaries of the results, and whatever other tools you may find useful.
    - Nearest neighbor
    - Bayesian Classifier
    - Decision Trees
    - Tree ensembles
    - Support Vector Machines
    - Neural Network Classifier
    - Comparison
5. Evaluation and Main Conclusions
    - What is the best model and the recommended data science procedure for the business?
    - What do you think that the business can gain from your data science effort?
    - What are the lessons learnt?
    - What is your summary of the achieved results?

To submit:

- a **fully operational Jupyter notebook** with the selected experiments as clear and concise as possible. Avoid output dumps.
- an **export of the jupyter notebook in html** format.

Recall that the report is going to be evaluated by your very busy professors and that they may have to skip many pages if your report is too long. Always highlight your best results. Please note:

- The objectives for each experiment and plot should be clear so that the reader understands why it is worth to read a particular part. Describe the goals of each figure and the main conclusions that you can obtain from the figure.
- The conclusion should be a short high level account of what was observed.
- It is **not necessary to describe the methods** (unless requested, but you should know their concepts and how they work). It is more important to point out the differences in the methods and the reasons for the results in terms of methods characteristics.
- A **4** minute video (or link to a video), per element of the group with a recorded presentation of the respective part of the work. The presentations of the group, when combined, describe the whole of the group's work and should not exceed the 12 minutes.
- The project slides presentation.

Kaggle

Go to the Kaggle platform and create an account for your group. The group should be named as the name of the group in the Moodle. Note that without this name I won't be able to consider your submissions.

The competition is hosted at the website: https://www.kaggle.com/t/3cfa310942d4425c947467d71f12d05e

And will be open until (17th of December): 17/12/2024 11:55 PM

After that I will communicate the results.

When you submit your predictions, during competition you will only see the results for half of the cases. The results for the remaining cases will be presented after the competition closes.

The score to evaluate the model and score the teams is the F1 score.

Evaluation

- This assignment is worth the values described in sigarra, according to the course you are following.
- Components
  - Report 30%
    - Narrative 10%
    - Writing style 10%
    - Presentation 10%
  - Technical 70%
    - Diversity of the results for the experiments 20%
    - Correctness 30%
    - Competition performance 10%
    - Conclusions 10%

Groups

Assignments are submitted by **groups of 3 students**. Different elements may have different grades. Other group sizes will not be considered.

It is advisable that the students from the same group perform overlapping work and only after that, exchange ideas with each other. Group work is important for learning from other people.

Submissions

Formal final deadline is **December 17th 2023 at 23:55**, to be submitted in Moodle, and only in Moodle. Submissions after that date will be multiplied by a monotonously decreasing factor that starts in 1.

- Checkpoints:

In the classes of the **19th of November** there will a checkpoint. Each group should present an update with the status of the project. You will have around 3/4 minutes for this presentation, where you can show your current results.

Ethical principles

When submitting, students commit themselves to follow strong ethical principles. All the work must be done by the elements of the group alone. All members of the group will be involved with the whole of the work. All the materials used and consulted must be credited in the work.