Homework Assignment 1

Predictive Analytics

2IID0 Web analytics
Submission deadline: November 28, 23:59 CET

Overview. In this assignment you will get some experience with predictive analytics. In the first part, you will perform some basic data exploration to understand the data. In the second part, you will build machine learning, in particular classification, models and assess their quality. You will learn how to deal with two common problems in predictive analytics: overfitting and class imbalance.

Dataset

In this homework assignment, you are given a data set about Speed Dating. In practice, data preprocessing is a big part of data science. For this assignment, we have been so kind to perform most of the data preprocessing for you, allowing you to focus on predictive analytics.

The original dataset was compiled by Columbia Business School professors Ray Fisman and Sheena Iyengar. Data was gathered from participants in experimental speed dating events from 2002-2004.

During the events, the attendees would have a four minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests. The dataset also includes questionnaire data gathered from participants at different points in the process. These fields include: demographics, dating habits, self-perception across key attributes, beliefs on what others find valuable in a mate, and lifestyle information.

The variables in the dataset can be divided into features and a target variable. The *target variable* is the variable we will try to predict. In the speed dating dataset, the target variable is the variable that indicates whether there exists a match or not. The *features* are all other variables that we will use to predict the target variable.

The dataset can be found at `canvas.tue.nl` under the name `speed_dating_assignment.csv`.

Tooling

You can use your own preferred programming language or machine learning environment for the assignments, e.g. Weka, RapidMinder, scikit-learn (Python) or R. For the data exploration exercises, Excel can be useful as well.

Submission

Submit your report as a PDF (with explicit answers to all questions) and code via `canvas.tue.nl`, before **November 28, 23:59 CET**.

Grading

The total number of points that can be earned with this assignment is 100. Your grade is equal to the number of points you achieved, divided by 10. A grade will be given to the group as a whole. **Please include a peer-review statement indicating whether everybody has contributed equally.** If someone contributed much more or much less than the others, please state this explicitly.

N.B.: Even if a question asks you to write code, it is imperative that you show us the logic behind the choices that you make in your report. Show that you understand your results by providing interpretations and drawing conclusions. Additionally, make sure that you give dedicated answers to all questions asked. Finally, we can only award points to answers that we understand, so make sure you pay attention to the readability of your report!

1 Data Exploration (20 points)

To get a better idea of the data that we are dealing with, we will start with some data exploration. We first investigate the distribution of each of the variables in the dataset.

a. (10 points) For each of the numerical variables in the dataset, compute the *mean* and *standard deviation*. For each of the non-numerical variables in the dataset, determine the *mode*. Additionally, for each of the variables, create a *histogram* and inspect it. Which variables are distributed as expected? Are there any variables with a very surprising distribution? Report the computed statistics and comment on the most interesting observations. Additionally, report the five most interesting findings inferred from the histograms.

Note. You don't have to include all histograms in the report, only those that provided interesting insights.

We now investigate how features in the dataset are related to each other. A useful statistic for this purpose is *correlation*. The correlation between two variables measures their linear relationship.

b. (10 points) Compute the correlations between all numerical variables in the dataset. Are there any surprising relationships? Can you identify features that are correlated to the target variable? Report all correlations you've computed (e.g. in a *correlation matrix*) and comment on your most interesting findings.

Hint. There exist many tools that allow you to compute correlations or correlation matrices, a.o. the `CORREL()` function in Excel, the `cor()` function in R, and the `corr()` function of the `pandas` package in Python.

2 Classification (80 points)

Now lets see if we can predict whether a date will be successful or not using machine learning. We will also deal with common problems in machine learning: *overfitting* and *class imbalance*.

Overfitting and underfitting

An important concept in machine learning is distinguishing between *training data* and *test data*. Training data is used to train your machine learning model, whereas testing data is used to test your machine learning model.

Splitting your data into a training and test set is important to ascertain that your model does not show signs of *overfitting*. Overfitting occurs when your model does not learn general patterns, but artifacts specific to the dataset it was trained on. For example, consider an unsophisticated model that checks whether an input instance corresponds to any of the instances it has seen before. If it does, the model returns the target value of that training instance. If it does not recognize the instance, it will simply return a random prediction. This model will be able to perfectly predict instances of the training dataset, but will be of little use for new data. In other words, this model does not *generalize* well to new data. We can check whether overfitting occurs by evaluating models based on the test set score rather than the training set score.

Underfitting, on the other hand, occurs when your model did not capture the underlying relationships in the data. For example, a linear model will never be able to capture non-linear relationships.

a. (40 points) Split the dataset into a training and test set (e.g. 75% of the instances for training and 25% of the instances for testing). Choose which features you will use in your models (e.g. based on your analysis in the previous exercise). Train at least three different machine learning models. You can choose from, for example, one of these algorithms: decision tree, k-nearest neighbor, naive Bayes, support vector machine, and random forest. Retrieve the *accuracy* of your models on both the test and training dataset. Accuracy is the fraction of instances the model predicted correctly. Which model performed best? Can you think of any reasons why a certain algorithm returns better results than another? Are there any signs of overfitting or underfitting?

In your report, describe which algorithms you used and with which settings. Briefly explain the intuition behind the algorithms that you used. Don't forget to reference any articles/books/etc. you used to understand the intuition behind the algorithm. Additionally, describe which features you included in the model, which ones you left out, and why. For each of your models, report both the training set accuracy and the test set accuracy and comment on the differences.

Class Imbalance

As you may have noticed in the first exercise of this assignment *(if not, you might want to go back to that exercise)*, only a small fraction of the dates resulted in a match. This is an issue that is referred to as *imbalanced data*. Accuracy does not take into account class imbalances, which makes it a less appropriate measure for imbalanced data (*can you reason why?*). Even if data is not imbalanced, we might associate different costs with different types of mistakes. For example, in cancer diagnosis, one may argue that a false negative (predicting the patient does not have cancer even though she does) is worse than a false positive (predicting the patient does have cancer even though she doesn't).

Most machine learning algorithms do not output a final prediction (i.e. match or no match) but a probability (i.e. the probability that the date results in match). This is also referred to as *confidence*. To get a better idea of the trade-off between different types of mistakes, we can investigate the *ROC curve*. This curve plots the false positive rate against the true positive rate for different probability thresholds. An associated performance measure is the *area under the ROC curve* (or AUC). Another performance measure that takes into account class imbalance is the *F1 score*.

There exist several strategies to deal with imbalanced data. One strategy is to give *different weights* to mistakes for different classes in the cost function of the algorithm. That is, you can give higher weight to mistakes in the minority class compared to mistakes in the majority class. Another strategy is to *resample* your data. You can either undersample the majority class or oversample the minority class.

b. (40 points) Choose one (or more) of the strategies to deal with imbalanced data. Rebuild your models using the chosen strategy. Investigate the ROC curve of your model and compute the AUC and/or F1 score. How do your previous models perform? How did the chosen strategy affect the performance of your models? In your report, describe which strategy you have chosen and how the strategy could help to deal with class imbalance. Report the performance scores on the test set of all models you built (including the ones you've built in exercise *2a*). Comment on the results.

Hint 1. Don't worry if your chosen strategy does not improve the performance of (all of) your models. As long as you show that you understand what you did and correctly reflect on the results, you can still receive full points for this question.
Hint 2. A tutorial on cost sensitive learning with WEKA can be found in the WEKA MOOC course parts 4.5 and 4.6 (see links on the next page).

– Tutorial: https://www.cs.waikato.ac.nz/ml/weka/mooc/moredataminingwithweka/slides/Class4-MoreDataMiningWithWeka-2014.pdf
– Video class 4.5: https://www.youtube.com/watch?v=N3R_Z8OlrIg
– Video class 4.6: https://www.youtube.com/watch?v=l9muPldOG30