Using Machine Learning Tools 2020
Assignment 1: Wine Quality

1 Overview

In this assignment, you will apply some popular machine learning techniques to the problem of predicting wine quality (as rated by expert tasters). A data set has been provided containing various chemical properties of many Portuguese wines and their corresponding quality.

The main aims of the prac are:

- to practice using tools for loading and viewing data sets;
- to visualise data in several ways and check for common pitfalls;
- to plan a simple experiment and prepare the data accordingly;
- to run your experiment and to report and interpret your results clearly and concisely.

This assignment relates to the following ACS CBOK areas: abstraction, design, hardware and software, data and information, HCI and programming.

2 Instructions

While you are free to use whatever IDE you like to develop your code, your submission should be formatted as a Jupyter notebook that interleaves Python code with output, commentary and analysis. This assignment is divided into several tasks. Clearly label your solution to each task in your notebook and include the code, results and some text analysis.

None of the tasks in this assignment require writing more than a few lines of code! One goal of the assignment is explore `sklearn`, `pandas`, `matplotlib` and other libraries you will find useful throughout the course, so feel free to use the functions they provide.

Chapter 2 of the reference book is based on a similar workflow, so you may look there for some further background and ideas.

Task 1: Load and pre-process the data set (15%)

Download the data set from MyUni using the link provided on the assignment page. The original source for this data is the UCI Data Repository, but it is important you use the version hosted on MyUni as we have modified it for this assignment! Load the wine data set from the csv file and summarise it with appropriate pandas functions such as `info()` and `describe()`. Answer the following questions in your notebook:

- How many wines are included in the data set?
- Are all the wines unique, and if not, how many unique wines are there?
- Are there any missing or invalid values in the data?
- Is the scale of the different attributes approximately equal?

If necessary, perform some simple pre-processing to prepare the data for the next steps. Explain in your notebook what you have done and why.

Task 2: Visualise the data (15%)

1. To get an overall view of the data distribution, plot frequency histograms of each attribute in the data set. Display the histograms in your notebook and comment on what you observe.

2. We are trying to predict the attribute "quality" from some combination of the other attributes. Calculate the correlation of each other attribute with quality. Which 3 attributes are most correlated with quality? For each of these 3 attributes, display a scatter plot of that attribute against quality. Comment on what you observe.

Task 3: Predicting wine quality: baseline (15%)

The value we are trying to predict, quality, is an integer that takes on discrete values between 1 and 10. We will approach this as a regression problem: that is, we will pretend that quality is a real number whose value we want to predict. If desired, we can round the value of our prediction to the nearest integer afterwards.

We will use root mean square error (rmse) as our error function when evaluating our predictor:

$$rmse(h, y) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h_i^2 - y_i^2)}$$

where y_i is the actual quality value of the ith data point and h_i is the value predicted by our model.

To perform prediction, we first need to split our data set into train and test sets. To start, do this by randomly selecting 20% of the data as test data, and use the remaining 80% as training data. You can use the inbuilt function `train_test_split` in sklearn to do this. The test data will not be used until the final evaluation of each model.

Before we start training sophisticated models, it's useful to try a simple estimation method. This serves as a sanity check and provides a baseline performance against which to measure other models. Find the median quality value in your training set, and use that as the predicted value for every data point (ignoring all attributes).

Answer the following questions in your notebook:

1. What is the rmse for this predictor, when trained and tested on the *training* set? (Remember, the test set is not for use until later).

2. Why do we use regression rather than classification to predict quality? Would there be any advantages to treating this as a classification problem instead? Explain your answer.

Task 4: Predicting wine quality (20%)

Fit a *linear regression* model to your training data. Again, sklearn has inbuilt functions to enable this. To get an idea of how successful this model is, apply it to your *training* data and calculate the rmse.

Fit a *k nearest neighbour* regression model to your training data, for varying values of k. This model is also available in sklearn, so you can apply it as you did the linear regressor.

A slightly more powerful model is a *decision tree*. We will examine this model in detail later in the course, but for now, you can apply it in the same way you applied linear regression, using the DecisionTree regressor in sklearn.

Make sure your notebook includes:

1. The rmse obtained using each model on the training data, for each of the various settings you tried.
2. A plot of the results, showing some combination of actual quality, predicted quality and/or errors.
3. Some comment on how the errors compare and why you think this might be the case.

Task 5: Cross validation (20%)

At this point, we are still evaluating our models on the training data only. Achieving a low rmse on the the training data does not necessarily mean the model will perform well on the test data. For example, a powerful model with many parameters may overfit to the training data.

Cross validation lets us test our models by repeatedly splitting the training data into separate training and validation sets, and testing performance on "unseen" data in the validation set. Perform *10 fold cross validation* for each model. This splits the training set into 10 equal size subsets, and uses each in turn as the validation set while training a model with the other 9. You should therefore have 10 error values for each cross validation run.

1. Do some simple analysis of the error values obtained for each model in the previous task, and comment on what you find.
2. On this basis, what do you think is the most suitable model that you've tried?

Task 6: Free choice! (15%)

Your final task is to improve the performance of your best predictor so far in one of the following ways:

- A different prediction model (look at other models in sklearn documentation)
- Tune one of your existing models by experimenting with different parameter settings
- Further pre-process or filter the data to make it more amenable to prediction by your models (beyond simple cleaning, as in Task 1). For example, consider the range and distribution of each feature.

The choice is yours but you need to justify why you think this might help and the analysis you do to measure how well it has worked. The point of this question is to make you think about the problem so your reasoning is more important than the final result you achieve (in this case, anyway!).

Finally, report the performance of your best model on the test data set.

3 Assessment

Hand in your notebook as an `.ipynb` file via the MyUni page. Don't include the data in your submission, but make sure your code expects the data files to be in the same directory as the notebook. Make sure your notebook includes your code and formatted (Markdown) text blocks explaining what you've done. Your mark will be based on both code correctness and the quality of your comments and analysis.

The prac is worth 25% of your overall mark for the course. It is due on Monday March 30 at 11.59pm.

Anthony Dick
February 2020