COSC 2673 Machine Learning

Assignment 1

Introductory Machine Learning

Learning outcomes:

- Develop familiarity with solving a machine learning task on a real dataset
- Practice loading and analysing the data
- Practice setting up evaluation framework to compare different approaches
- Develop written communication skills to describe approach taken, explaining the reasoning for it and present results to justify the choices.

Group size: 1 (individual)

Due Date:

- Completed Assignment: 11.59pm Sunday, 26th August 2018

Introduction

In this assignment, you'll apply machine learning techniques and approaches to a regression problem. The assignment is aimed to develop your familiarity with applying machine learning techniques to learning problems, and to participate in a Kaggle in-class competition to compete with your class mates and get bragging rights (and bonus marks).

Task

Prediction is an important aspect of machine learning and data science in general. For this assignment, you'll be predicting the cancer death rate of some local regions in the US. The dataset consists of a number of features relating to the demographics of these regions, and the task is to use these to train a regression approach to predict the cancer death rate on test and unseen data.

You will also setup the evaluation framework, including selecting appropriate performance measures and determining how to split the data into training and testing.

As one of the aims of the assignment is to get you familiar with the machine learning paradigm, you should also evaluate a few different regression algorithms to determine which one would be most appropriate to predict the death rates.

Data

The data is available as part of the assignment download (see Canvas). We have cleaned it up for you, such that all the attributes/features are integers or floats, and missing values has been estimated and filled in. There are the following files:

- train.csv, contains the training dataset. Use this for both training your Kaggle submission and for your own exploration and evaluation of which approach you think is "best" for this prediction task.

- test.csv, containts the testing dataset for the Kaggle competition. It has all the independent features but not the dependent one (TARGET_deathRate). For Kaggle, you predict the values for the unknown TARGET_deathRate and submit them. There will be more details about this within the Kaggle competition page. For your own evaluation etc, this file may be useful for exploring the features.
- The file metadata.csv contains some brief description of each of the fields.

The original data is from https://data.world/nrippner/ols-regression-challenge, and the data we provided is based on this, with some perturbation. Online, there are one or two scripts and kernels available for this dataset and have found the target feature is strongly correlated with several independent features. As the aim of this assignment is to encourage you to learn to setup evaluation and explore different approaches, your attempted approaches must not explicitly perform feature selection, i.e., your learning models should have all features as input.

Kaggle in-class Competition

In addition to doing your own evaluations (strongly recommended), we would like you to enter the Kaggle in-class competition we have setup for this.

The location of the competition will be released with about 2 weeks to go, as we wish you to first explore with your own code before entering Kaggle, which can be dangerous for beginning machine learning students (real machine learning is not about optimising ones model to fit the test data, which is what sometimes Kaggle competitions boils down to). However, we do want you to explore and we are using Kaggle as an extra incentive for that.

The top 3 submissions will have bonus marks given to the individuals – 5 bonus marks for first, 3 bonus marks for second and 1 bonus mark for third.

To be able to evaluate this properly, we need you to use an account name that either is your full name or your student number. If you have a Kaggle account already and it isn’t one of the above, please create a new one for this in-class competition. If you haven’t got one, please create one, and hopefully this in-class competition will introduce you to the interesting and exciting competitions that occur on Kaggle.

Once created, log onto Kaggle and provided URL (to be provided, as described above). Within Kaggle there are more details, but essentially you will be predicting the unknown values for test.csv, and your position on the leader board will be based on how well you perform. There is a limit of 2 submissions per day, as we want to encourage you to develop off-line and do your own evaluation first before submitting periodically to Kaggle.

Report

In addition to the Kaggle in-class competition, part of your assessment is based on a (up to) 7 A4 page report, at least in font size 12. In this report you’ll describe your final selected approach, why you selected this approach, parameter settings and some other regression approaches you have tried. This will allow us to understand your rationale, but also encourage exploration and not just focused on maximising a single performance metric (i.e., the Kaggle in-class).

Getting Started

To help you get started, we suggest the following:

- Load dataset into your Jupyter or your favourite Python IDE
- Do some preliminary data exploration, to understand it better (this will help you later on with trying to figure which regression approach is ideal and how to improve it)
- Setup your data into training and testing datasets
- Select the basic linear regression algorithm and train it then evaluate it
- Analyse the results and see what is going on (to help you determine what needs to be changed to improve the regression model)

Sources of Help

Your lecturers would be very happy to discuss questions and your results with you. Please feel free to come talk to us during consultation, or even a quick question, during lecture break. Use the existing communication channels you have been using with us also.

Also you can ask questions on Canvas, but please do not post any code.

There will also be a FAQ, and anything in the FAQ will override what is specified in this specifications, if there is ambiguity.

Plagiarism

Remember to gain maximum benefit from our time together in this course, try things yourself. See https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity for details about academic integrity.

This including asking or paying someone else to do your assignment. We do care about fair outcomes for all students.

Assessment

The assessment is based on your report, your code and a bonus based on your final position in Kaggle.

Examine the assessment rubric below for more details.

At a high level, you'll be assessed on how well you communicated your work, your approach, your explanation and justification for it. As a rough rule of thumb, you can get a credit and distinction if you just reiterate the material learnt in labs, but to get a HD would generally require you doing some research, exploration and going beyond what we did in class. More will be described in lectures.

What to submit

- Your report, up to 7 A4 pages in length and in font size 12 of assessed content, not including any cover page or appendices. Note this is a maximum, not a length you need to must have.

Anything beyond 7 pages will not be read, and anything in appendix should be considered as additional information, as there is no guarantee it will be read. Please do not ask for more pages, this limit is strict.
- Your Python scripts or Jupyter notebooks used to perform your analysis. Please comment and following good practices in regards to the code (e.g., reasonable variable names, no magic numbers etc) as that will form part of the assessment.
- Submit at least one submission on the Kaggle in-class competition for possible bonus.

Submission should be made to Canvas. Closer to submission we will describe the submission process in more detail.

Rubric

Use the following rubric to help you determine how to approach the assignment.

Criteria	Excellent	Good	Fair	Poor
Code Style and Readability (20%)	Code is styled and organised well, following general good programming practices. It is well commented and easy to follow the logic.	Code is styled and organised reasonably, following general good programming practices. Commenting could be improved, but generally possible to follow logic after some work.	Code is styled and organised poorly, not following general good programming practices. Commenting is rare.	Code is styled and organised poorly, not following general good programming practices. Commenting is absent.
Approach (60%)	The approach is an appropriate method to take to solve the problem. Approach taken goes beyond using the tools provided in class. Submission justifies and explains the approach well. Approach includes techniques evaluated, training and evaluation setup and justification of parameter settings.	The approach is an appropriate method to take to solve the problem. Approach taken is limited to mostly the techniques discussed in class. Submission justifies and explains the approach well. Approach includes techniques evaluated, training and evaluation setup and justification of parameter settings.	There are other approaches that are clearly more appropriate method to take to solve the problem. Approach taken is limited to the tools provided in class. Submission somewhat justifies and explains the approach, but there are unexplained choices. Approach includes data collected and techniques used.	There are other approaches that are clearly more appropriate method to take to solve the problem. Approach taken is limited to the tools provided in class. Submission does not justifies and explains the approach. Approach includes data collected and techniques used.
Report Presentation	Report is easy to read and flows	Report is reasonably easy to	Report is difficult to follow in places	Report is difficult to follow and

Criteria	Excellent	Good	Fair	Poor
(20%)	well. It is structured well, leading the reader to fully understand the rationale for the final approach taken. Approaches are described well, training and testing setup is described, and parameter settings are justified. Tables, figures and other visualisation are easy to read and to interpret.	read and flows relatively well. It is structured reasonably well, leading the reader to reasonably understand the rationale for the final approach taken. Approaches are described well, training and testing setup is described, and parameter settings are mostly justified. Tables, figures and other visualisation are easy to read and to interpret.	and doesn't flows well. It is adequately structured, but reader may find it difficult to understand the rationale of selected approach. Approaches are described but not in much detail, training and testing setup may be described, and parameter settings are not really justified. Tables, figures and other visualisation are either too small or difficult to interpret.	doesn't flows well. It is barely structured and readers find it difficult to understand the rationale of the selected approach. Approaches are barely described, training and testing setup is likely to be missing, and parameter settings are not justified. Tables, figures and other visualisation are either too small or difficult to interpret.