Practical 1

Dr Simon Denman
CAB420: Machine Learning

This weeks practical will focus on linear regression, and data manipulation; and will also serve as a Python refresher. This practical uses data in the `CAB420_Prac1.zip` file available alongside this document.

Question 1 is optional and deals with general data wrangling and pre-processing. This is not a focus of CAB420, yet nonetheless is an important skill to develop and something that will likely encounter in the group assignment (and elsewhere when dealing with real-world data). Students are encouraged to tackle both questions; and the combined dataset that results from Question 1 is used in Question 2. For those that wish to focus on the regression aspects first (Question 2), a pre-combined data set (`combined.csv`) is provided.

Problem 1. Combining and Filtering Multiple Datasets. `CAB420_Prac1.zip` contains a number of datasets, split into two directories as follows:

1. `BOM`, which contains Bureau of Meteorology data for Brisbane City from the years 1999-2019. The data is split into three files.
   (a) `IDCJAC0009_040913_1800_Data.csv` contains daily rainfall data;
   (b) `IDCJAC0010_040913_1800_Data.csv` contains maximum daily temperature data; and
   (c) `IDCJAC0013_040913_1800_Data.csv` contains daily solar exposure data.
2. `BCCCyclewayCounts` contains five years data (from 2014-2018) for Brisbane City Council cycleways, with data for each year being in a separate file (i.e. `bike-ped-auto-counts-2014.csv` contains data for the year 2014).

You are to combine these datasets into a single table using Python (or the programming language of your choice) such that:

- You have a single table that spans the time period of the `BCCCyclewayCounts` data;
- Duplicate information is avoided (i.e. you don't have multiple date columns, or similar);
- For the cycle way data, only columns that are available in all years data are included in the final table (i.e. if a counter is available in $2014 - 2017$, but not 2018, that column should be excluded.

There are many ways to approach this task (particularly in python). You may use any approach you see fit, but a suggested approach would be:

1. Load all the BOM data. Identify what columns are common between the tables (for the BOM data, this could be done by manual inspection or through code); and which ones are unique. Merge the tables such that one table contains all columns, without duplicate columns (you may merge tables and then remove duplicates; or copy only relevant columns to a combined table).

2. Load the BCC tables. Identify which columns are common between the tables. You could consider treating the columns from each file as a set and testing for set membership between the columns from the different files to achieve this (note: identifying common columns manually will be time consuming, a programmatic approach is recommended).

3. Remove dates from the BOM data that are not in the BCC data, ensuring that the two sets of date are time-aligned (i.e. rows in the two tables correspond to the same date).

4. Merge the BOM and BCC data.

For python users, you may find the `pandas` package of use. In particular, you may wish to use:

- `pandas.read_csv`: This will load a CSV file into a pandas data frame
- `pandas.to_datetime` and `pandas.apply`: This will help create consistent date objects for the BCC and BOM data. For the BOM data, a suggested approach is to create your own function to create a date from individual columns, and use `pandas.apply`.
- `pandas.concat`: A function to concatenate pandas table.
- `pandas.merge`: A function to combine two pandas tables, using a particular column as the key.

Problem 2. Linear Regression. Using the dataset from **Problem 1** (or the `combined.csv` merged dataset), split the data into training, validation and testing as follows:

- Training: All data from the years 2014-2016
- Validation: All data from 2017
- Testing: All data from 2018

Develop a regression model to predict one of the cycleway data series (select whichever one takes your fancy) in your dataset. In developing this model you should:

- Initially, use all weather data (temperature, rainfall and solar exposure) and all other data series for a particular counter type (i.e. if you're predicting cyclists inbound for a counter, use all other cyclist inbound counters).
- Use p-values, qqplots, correlation between predictors and response, correlation between pairs of predictor, sand performance on the validation set to remove terms and improve the model.

When you have finished refining the model, evaluate it on test set, and compare the Root Mean Squared Error (RMSE) for the training, validation and test sets.

In training the model, you will need to ensure that you have no samples (i.e. rows) with missing data. As such, you should remove samples with missing data from the dataset before training and evaluating the model. This may also mean that have to remove some columns that contain large amounts of missing data (i.e. determine how many samples are missing in each column, remove columns with lots of missing data, remove any other rows where data is missing).

For python users, you may find the `statsmodels` and `pandas` packages of use. In particular, you may wish to use:

- `isna`: A member function of a pandas dataframe that indicates if a variable is missing.
- `dropna`: A member function of a pandas dataframe that drops missing values.
- `statsmodels.api.OLS`: Ordinary Least Squares regression function within `statsmodels`.

Python users may also wish to explore the `sklearn` package which also contains methods for linear regression, and data splitting (we will be using `sklearn` next week).

For students who do not wish to complex question 1 before moving onto question 2, a merged dataset is provided in `combined.csv` on blackboard.