Assignment 3

Practical Introduction to Data Science (SS1-SEM2)

Candidate Exam Number: B136325

Department of Informatics
University of Edinburgh
Scotland
9th June 2019

Contents

Question 1 **1**
Introduction . . . 1
Business Understanding . . . 1
GitHub . . . 1
RStudio . . . 2
Data understanding . . . 2
Statistical value chain . . . 2
Raw data . . . 2
Technically correct: weather .txt data . . . 2
Technically correct: weather .dsv data . . . 3
Technically correct: weather data frames . . . 3
Technically correct: weather data frames correct . . . 3
Technically correct: weather single data frame . . . 4
Exploratory data analysis (EDS) . . . 4
Summary . . . 4
Box and whisker . . . 4
Remove outliers . . . 4
Weather variables by latitude . . . 4
Cluster tendency . . . 5
Algorithm choice . . . 5
Data preparation . . . 5
Determining K . . . 5
Distance metric . . . 5
Feature selection . . . 5
Analysis . . . 6
K-means . . . 6
Model based . . . 6
Evaluation . . . 6
Deployment . . . 6

Question 2 **7**

Question 3 **8**

Appendices **10**
Raw data output 11

Question 1

Introduction

The demonstration will be structured with regard to the CRISP-DM data science project methodology, which describes such projects in terms of the following six sequential phases: Business Understanding, Data Understanding, Data Preparation, Analysis, Evaluation and Deployment. The first of those phases, Business Understanding, will be presented immediately below.

Business Understanding

The goal of this question will be to demonstrate that weather data can be clustered into two or more groups of 'similar' weather conditions, using a machine learning clustering algorithm. The weather data will contain the results from thirty six weather stations across the UK (as made available online by the Met Office). While the process of retrieving and then cleaning the data will be described within the Data Understanding stage further below, it is worth noting that all of the process described within this answer have been implemented as an R package called *pids-wellbeing-weather*. Moreover, all of the processes have been exported from the package as named functions.

GitHub

The *pids-wellbeing-weather* package has been made available on GitHub, using this student's exam number, b136325, as the GitHub account name. In addition, the package makes use of Packrat dependency management. This mean that the packages contains all of its dependencies. This approach increases the size of the package. Importantly, however, it ensures portability. The package can be cloned from GitHub using the following command:

```
git clone https://github.com/b136325/pids.wellbeing.weather.git
```

RStudio

Alternatively, the package can be installed into RStudio using the commands below:

```
install.packages("devtools")
library(devtools)
install_github("b136325/pids.wellbeing.weather")
```

Data understanding

Statistical value chain

1. Package directory structure - raw to data
    2. Within data (svc) sub-directories
    3. sub directory naming
    4. Command naming
    5. single exported command per file

Raw data: weather and well-being

This process downloads the required files into the *pids-wellbeing-weather* and stores them within the data-raw/weather and the data-raw/wellbeing directories, respectively. The web address from which the data are downloaded are constructed dynamically using constants defined within 'constants-weather-station-names.R' and 'contants-wellbeing'. The function associated with this process has been exported from the pids.wellbeing.weather package. When run it returns a list of files that have been downloaded successfully, along with any that have failed, as illustrated in Figure **??**. The function can be run using the command below:

```
pids.wellbeing.weather::question_1_001_raw_data()
```

Technically correct: weather .txt data

In order to ensure a strong data related 'separation of concerns', such that, no amendments should be made to the data stored within the 'data-raw' directory, this process moves the raw weather data from 'data-raw' into 'data'. In addition, it adds a closing new line character to each file, facilitating subsequent processing. More specifically, the weather data is moved into a directory representing the first stage of the 'technically correct' Statistical Value Chain process, which can be found at 'data/weather/stage-010-technically-correct-text'. The function associated with this process has been exported from the package and it can be run using the command below. The function returns a list containing the destination paths of the successfully moved files.

```
pids.wellbeing.weather::question_1_002_tech_correct_weather_txt()
```

Technically correct: weather .dsv data

The next stage of technically correct data processing transforms the .txt files into a .dsv white space delimited file format. This processes involves the removal of non column related header items from the .txt files. It also involves ensuring that the all of the files have a standard number of columns, which enables them to be reliably converted into a delimited file format. In addition, invalid data items, such as "—" are converted into a common data strandard, such as "NA". The function associated with this process has been exported from the package and it can be run using the command below. The function returns a list containing the destination paths of the successfully moved files, along with any errors.

```
pids.wellbeing.weather::question_1_003_tech_correct_weather_dsv()
```

Technically correct: weather data frames

The stage of technically correct data processing transforms the .dsv files into data frames, saved in .Rds formats within the data/weather/stage-012-technically-correct-dataframe. The function associated with this process has been exported from the package and it can be run using the command below. The function returns a list containing the destination paths of the successfully saved files, along with any errors.

```
pids.wellbeing.weather::question_1_004_tech_correct_weather_df()
```

Technically correct: weather data frames correct

The stage of technically correct data processing transforms the .Rds files from the data/weather/stage-012-technically-correct-dataframe directory. It ensures that the data frames have common lower case column names, along with appropriate data types. The transformed data frames are stored within the 'stage-013-technically-correct-complete' directory. The function associated with this process has been exported from the package and it can be run using the command below. The function returns a list containing the destination paths of the successfully saved files, along with any errors.

```
pids.wellbeing.weather::question_1_005_tech_corr_weather_complete()
```

Technically correct: weather single data frame

The stage of technically correct data processing transforms the .Rds files from the ‘stage-013-technically-correct-complete‘ directory into a single data frame, which is stored within ‘stage-014-technically-complete-single-dataframe‘. The function associated with this process has been exported from the package and it can be run using the command below. The function returns the destination path of the successfully saved file, along with any errors.

```
pids.wellbeing.weather::question_1_006_tech_corr_weather_single_df()
```

Exploratory data analysis (EDA)

Summary

Summary Type	Sunshine (hours)	Rain (mm)	Temp max (c)	Temp min (c)
Mean	119.119	75.032	12.816	5.998
Min	89.526	45.803	9.496	2.734
Max	155.310	151.117	14.968	8.386
SD	16.196	26.958	1.343	1.194

Table 1: Mean, min, max and SD for the weather features (3sf)

```
pids.wellbeing.weather::question_1_007_grouped_data()
pids.wellbeing.weather::question_1_008_grouped_data_summary()
```

Box and whisker

```
pids.wellbeing.weather::question_1_009_eda_charts_box_whisker_hours_sun()
pids.wellbeing.weather::question_1_009_eda_charts_box_whisker_hours_rain()
pids.wellbeing.weather::question_1_009_eda_charts_box_whisker_max_temp()
pids.wellbeing.weather::question_1_009_eda_charts_box_whisker_min_temp()
```

Remove outliers

```
pids.wellbeing.weather::question_1_010_remove_outliers()
pids.wellbeing.weather::question_1_011_remove_outliers_summary()
```

Weather variables by latitude

```
pids.wellbeing.weather::question_1_012_eda_charts_latitude()
```

Summary Type	Sunshine (hours)	Rain (mm)	Temp max (c)	Temp min (c)
Mean	118.43	69.602	12.816	5.998
Min	91.320	47.399	9.496	2.734
Max	149.349	127.368	14.968	8.386
SD	14.032	18.998	1.343	1.194

Table 2: Mean, min, max and SD for the weather features (3sf)

```
pids.wellbeing.weather::question_1_013_eda_charts_hours_sun()
pids.wellbeing.weather::question_1_014_eda_charts_rain()
pids.wellbeing.weather::question_1_015_eda_charts_max_temp()
pids.wellbeing.weather::question_1_016_eda_charts_min_temp()
```

Cluster tendency

The clustering tendency of the data has been calculated using the Hopkins statistic (H). It assesses the probability that the data contains non random structures. The statistic has been calculated using the *factoextra* dependency. Using the data with outliers removed, the result of H was **0.352**.

(1) high score–¿uniform distribution–¿no cluster

(2)low score–¿not uniform distribution-/-¿(may be not)cluster.

The function used to generate this statistic can be run using the command below:

```
pids.wellbeing.weather::question_1_017_cluster_tendency()
pids.wellbeing.weather::question_1_017_cluster_tendency(show_chart = TRUE)
```

Algorithm choice

WHY USE A PARTITION METHOD RATHER THAN DENSITY

```
pids.wellbeing.weather::question_1_018_algorithm_choice()
```

Data preparation

Determining K

1˙019

Distance metric

Feature selection

1˙020

Analysis

k-means

```
pids.wellbeing.weather::question_1_021_k_means()
pids.wellbeing.weather::question-1-022-charts-sum-squares()
```

```
pids.wellbeing.weather::question_1_021_k_means()
pids.wellbeing.weather::question-1-022-charts-sum-squares()
```

1˙23 to 1˙26

Model based

Evaluation

Deployment

Question 2

Question 3

Bibliography

Appendices

A. Raw data output