FIT1043 Assignment 2: Specification

Due date: Monday 2nd October 2023 - 11:55 pm

Aim

The main objective of Assignment 2 is to conduct predictive analytics, by building predictive models on a dataset using Python in the Jupyter Notebook environment.

This assignment will test your ability to:

- Read and describe the data using basic statistics,
- Split the dataset into training and testing,
- Conduct multi-class classification using Support Vector Machine (SVM)**,
- Evaluate and compare predictive models,
- Explore different datasets and select a particular dataset that meets certain criteria
- Deal with missing data,
- Conduct clustering using k-means

** Not taught in this unit, you are to explore and elaborate these in your report submission. This will be a mild introduction to life-long learning to learn by yourself.

Data

We will explore the following datasets in **Part A** (plus a dataset of your choice in **Part B**):

1. FIT1043-Essay-Features.csv
2. FIT1043-Essay-Features-Submission.csv

Format: each file is a single comma separated (CSV) file
Description: These two datasets were derived from a set of essays and are used to describe the essay features in numeric information.
Columns:

Column's header	Description
essayid	a unique id to identify the essay
chars	number of characters in the essay, including spaces
words	number of words in the essay
commas	number of commas in the essay
apostrophes	number of apostrophes in the essay
punctuations	number of punctuations (other than commas, apostrophes, period, questions marks in the essay
avg_word_length	the average length of the words in the essay
sentences	number of sentences in the essay, determined by the period (fullstops)
questions	number of questions in the essay, determined by the question marks

avg_word_sentence	the average number of words in a sentence in the essay
POS	total number of Part-of-Speech discovered
POS/total_words	fraction of the POS in the total number of words in the essay
prompt_words	words that are related to the essay topic
prompt_words/total_words	fraction of the prompt words in the total number of words in the essay
synonym_words	words that are synonymous
synonym_words/ total_words	fraction of the synonymous words in the total number of words in the essay
unstemmed	number of words that were not stemmed in the essay
stemmed	number of words that were stemmed (cut to the based word) in the essay
score	the rating grade, ranging from 1 – 6

This data is pre-processed data on a set of essays that were provided on Kaggle. You DO NOT have to download or process/wrangle the data from the original source.

Hand-in Requirements

Please hand in the following 4 files (including a **PDF file** containing your code, answers and explanations to questions, a **Jupyter notebook file (.ipynb)** containing your Python code to all the questions, **CSV file** for your prediction in task A4 and the **video file** respectively):

- The PDF file should contain:
  - Answers and explanations to the questions. Make sure to include screenshots/images of the graphs you generat. Also, copy/paste your Python code to **justify your answers** for all the questions.
  - You can use Microsoft Word or other word processing software to format your submission. Alternatively, generate your PDF from your jupyter notebook formatted using markdown. Either way save the final copy to a PDF before submitting.

- The .ipynb file should contain:
  - **A copy of your work using python code** to answer all the questions.

- The video file should contain:
  - **A recording of yourself, explaining your answers to Part B.**
  - You can use Zoom to prepare your recording.
  - Note each student is required to explain their approach in Part B only. Please see Part B for more details.
- A csv file of your predictions in A4

Note: Zip, rar or any other similar file compression format **is not acceptable** and will have a **penalty of 10%**.

Assignment Tasks:

Note: You need to use Python to complete all tasks.

Part A: Classification

A1. Supervised Learning

1. Explain supervised machine learning, the notion of labelled data, and train and test datasets.
2. Read the '**FIT1043-Essay-Features.csv**' file and separate the features and the label (Hint: the label, in this case, is the 'score')
3. Use the sklearn.model_selection.train_test_split function to split your data for training and testing.

A2. Classification (training)

1. Explain the difference between binary and multi-class classification.
2. In preparation for classification, your data should be normalised/scaled.
    a. Describe what you understand from this need to normalise data (this is in your Week 7 applied session).
    b. Choose and use the appropriate normalisation functions available in sklearn.preprocessing and scale the data appropriately.
3. Use the Support Vector Machine algorithm to build the model.
    a. Describe SVM. Again, this is not in your lecture content, you need to do some self-learning.
    b. In SVM, there is something called the kernel. Explain what you understand from it.
    c. Write the code to build a predictive SVM model using your training dataset.
    (Note: You are allowed to engineer or remove features as you deem appropriate)
4. Repeat **Task A2.3.c** by using another classification algorithm such as Decision Tree or Random Forest algorithms instead of SVM.

A3. Classification (prediction)

1. Using the testing dataset you created in **Task A1.3** above, conduct the prediction for the 'score' (label) using the two models built by SVM and your other classification algorithm in A.2.4.
2. Display the confusion matrices for both models (it should look like a 6x6 matrix). Unlike the lectures, where it is just a 2x2, you are now introduced to a multi-class classification problem setting.
3. Compare the performance of SVM and your other classifier and provide your justification of which one performed better.

A4. Independent evaluation (Competition )

1. Read the '**FIT1043-Essay-Features-Submission.csv**' file and use the best model you built earlier to predict the 'score' for the essays in this file.
2. Unlike the previous section in which you have a testing dataset where you know the 'score' and will be able to test for the accuracy, in this part, you don't have a 'score' and you have to predict it and submit the predictions along with other required submission files.
    a. Output of your predictions should be submitted in a CSV file format. It should contain 2 columns: 'essayid' and 'score'. It should have a total of 200 lines (1 header, and 199 entries).

Part B: Selection of Dataset, Clustering and Video Preparation

B1. Selection of a Dataset with missing data, Clustering

We have demonstrated a k-means clustering algorithm in week 7. Your task in this part is to find an interesting dataset and apply k-means clustering on it using Python. For instance, Kaggle is a private company which runs data science competitions and provides a list of their publicly available datasets: https://www.kaggle.com/datasets

1. Select a suitable dataset **that contains some missing data and at least two numerical features**. Please **note** you cannot use the same data set used in the applied sessions/lectures in this unit. Please include a link to your dataset in your report. You may wish to:
    - provide the direct link to the public dataset from the internet, or
    - place the data file in your Monash student - google drive and provide its link in the submission.
2. Perform wrangling on the dataset to handle the missing data and explain your procedure
3. Perform k-means clustering, choosing two numerical features in your dataset, and apply k-means clustering to your data to create k clusters in Python (k>=2)
4. Visualise the data as well as the results of the k-means clustering, and describe your findings about the identified clusters.

B2. Video Preparation

Presentation is one of the important steps in a data science process. In this task you will need to prepare a short video of yourself (you can share your code on screen) and describe your approach on the above task (**Task B1**).

- Please make sure to keep your camera on (show yourself) during recording. You may want to share your screen with your code while you talk.)

Good Luck! ☺