Assessment Submission Cover Sheet

This Assessment Cover Sheet **must** be included on all Assessment submissions.

|  |  |
| --- | --- |
| Assignment Title | CA2 – Text Mining |
| Module | Data Mining |
| Student Name  (same as Student Card) | Ciaran Finnegan |
| Student Number | D21124026t |
| Programme |  |
| Part-Time/Full-Time |  |
| Year of Study  (First Year, Second Year, etc) |  |

Late Submissions: Assessment submitted after the deadline will have a late penalty applied.

**Academic Integrity for assessment in TU Dublin Programmes**

Each student is responsible for knowing and abiding by TU Dublin Academic Regulations and Policies. Any student in breach of these regulation/policies will be subject to action in accordance with the University’s procedures for breaches of assessment regulations. Please refer to the General Assessment Regulations at

<https://tudublin.libguides.com/c.php?g=674049&p=4794713>

<https://www.tudublinsu.ie/advice/exams/breachesofregulations/>

All students are expected to complete their courses/programmes in compliance with University regulations. No student shall engage in any activity that involves attempting to receive a grade by means other than honest effort, for example:

1. No student shall complete, in part or in total, any examination or assessment for another person.
2. No student shall knowingly allow any examination or assessment to be completed, in part or in total, for themselves by another person.
3. No student shall plagiarise or copy the work of another and submit it as their own work.
4. No student shall falsify any data. Falsification is the invention of data, its alteration, its copying from any other source, or otherwise obtaining it by unfair means, or inventing quotations and/or references.
5. No student shall use aids or devices excluded by the lecturer in undertaking course work or assessments/ examinations.
6. No student shall knowingly procure, provide, or accept any materials that contain questions or answers to any examination or assessment to be given at a subsequent time.
7. No student shall provide their assignments, in part or in total, to any other student in current or future classes of this module/ programme unless authorised to do so by the lecturer.
8. No student shall submit substantially the same material in more than one module/programme without prior authorization.
9. No student shall alter graded assignments or examinations and then resubmit them for regrading, unless specifically authorised to do so by the lecturer.
10. All programming code and documentation, unless correctly referenced, submitted for assessment or existing in the student’s computer accounts must be the students’ original work or material specifically authorized by the lecturer.
11. Collaborating with other students to develop, complete or correct course work is limited to activities explicitly authorized by the lecturer.
12. For all group assignments, each member of the group is responsible for the academic integrity of the entire submission. Consequently, all group members must satisfy themselves that all elements of their submission adhere to the academic integrity statement points above.

By submitting coursework, either physically or electronically, you are confirming that it is your own work (or, in the case of a group submission, that it is the result of joint work undertaken by members of the group that you represent) and that you have read and understand the University’s Regulations and Policies covering Academic Integrity (see General Assessment Regulations)*.*

Coursework may be submitted to an electronic detection system in order to help ascertain if any plagiarised material is present. If you have queries about what constitutes plagiarism, please speak to your lecturer.

|  |  |
| --- | --- |
| Student Signature |  |
| Date |  |

IMPORTANT:

* Complete the required number of tasks as defined in Assessment Handout
* The sections listed below are an example of the section headings for each task. You can use alternative headings
* Tasks 1-3: Sub-Sections 1-7 should be no longer than 8 pages (minimum 6 pages), including diagrams, images, screen captures, tables, etc. Careful selection of these is needed.
  + Code does not count to this total. Code should be added to the relevant section.
* Detailed discussion is expected. Marks are awarded based on depth of information given.
* Marks are awarded based on complexity of problem and depth of work.

# TASK 1 – *Text Mining – Comparison of Rotten Tomatoe Movie Reviews in Word Clouds*

1. **Definition of Problem**

Clearly state the problem definition, what type of data mining task is it, where was the data set sourced from, etc.

Rotten Tomatoes is a very well know movie database that allows critics and the general public (defined as ‘users’ in this assignment) to upload reviews and scores on new movie releases.

It has often been claimed that film critics are out of touch with the taste of regular movie goers and that this can be seen by the dichotomy between critic and user scores on Rotten Tomatoes, particularly for the larger commercial (‘blockbuster’) movie titles.

This assignment attempts to provide some data analytical rigour to this assertion of ‘aloofness’ by movie critics. Although one can just compare the scores given by critics and users to movies, this assignment attempts to identify noticeable differences in the patterns of the language used in the reviews themselves.

Review data from four of the most recent movies in the Marvel Cinematic Universe is pulled from the Rotten Tomatoes website, and Text Mining techniques are used to present opposing WordClouds built from the critic and user reviews.

Although it is only a sample of four movies, this Word Cloud analyse will provide a visual indication of the disparity, if any, between the reviews from critics and users.

1. **Data Exploration & Descriptive Analytics**

Include any data insights discovered

For simplicity, the assignment focuses on the Word Cloud analysis of the reviews for the Marvel movie ‘Eternals’. However, the Python code is build in a generic way to extract and analyse the review data from any given move on the Rotten Tomatoe website.

*Calling APIs*

For reasons explained in Section 6 of this section of the report, the data extraction from Rotten Tomatoes was carried out via the publicly available API.

Based on technical specifications and code snippets from StackOverflow[x], two functions were written to separately extract the critics and the users movie reviews.

<Critics function>

Fig x – Python Function to Extract Movie Reviews from Critics for ‘Eternals’

The function to extract the User reviews differs only in the API line.

<Code Segment for User Reviews>

Fig x – Code Snippet to Invoke APIS for User Reviews

Obviously, it would be better programming practice to have a single function to extract reviews and parse the API call using a flag in the function parameter. For a number of reasons, I kept the functions separate, despite the duplication in code:

* Simplicity. The primary focus of this assignment is intended to be on setting optimal parameters for the Word Cloud display an I wanted to move quickly to that stage of development.
* The User reviews are considerably larger in volume than the review data from the Critics and take a significant amount of time to extract, usually in the order of several minutes. I experimented with different ‘Sleep()’ values for both sets of review for ‘Eternals’.

*Working with Dataframes*

The output from the APIs populate dataframes with a collection of data attributes for a given movie.

The dataframes returned by the APIs differ in structure between Critics and Users.

<Image of code – head – and critics dataframe>

Fig x – Dataframe for Critics Reviews

<Image of code – head – and users dataframe>

Fig x – Dataframe for User Reviews

Running a simple line of code pulls out and concatenates all the review text from each dataframe.

The ‘quote’ column contains the text from Critic reviews for ‘Eternals’.

<Image of creating critics text>

Fig x – Extract and Concatenate Critic Reviews into Text Variable

The ‘review’ column contains the text from User reviews for ‘Eternals’.

<Image of creating users text>

Fig x – Extract and Concatenate User Reviews into Text Variable

These text variables then form the starting point of the data preparation phase in the following section.

1. **Data Preparation**

Include details of any data cleaning, transformations, data enrichment, feature engineering, feature reduction, etc

3 – Tokenize the Words

The next step is to Tokenize the text. This breaks the text into individual words.

4 – Filter words, Remove Numbers & Punctuation

There will be a lot of things in the text that we don’t want included in the analyse. We want the text to only contain words. The following extracts the words and ignores numbers, punctuation, etc.

5 – Setup Stop Words

Stop words are general words in a language that doesn’t contain any meanings and these can be removed from the data set. Python NLTK comes with a set of stop words defined for most languages.

Additional stop words can be added to this list. I added the words listed below. Some of these you might expect to be in the stop word list, others are to remove certain words that appeared in the various manifestos that don’t have a lot of meaning. I also added the name of the parties  and some Irish words to the stop words list.

Now remove these stop words from the list of tokens.

The number of tokens is reduced to 31,038

1. **Details of Algorithms & Configurations**

Building the WordCloud

1. **Model Performance Metrics & Evaluation of Results**
2. **Comparison with other Research & Reflections**

Compare your results to at least three other researchers (maximum of five) who used the same data set. What lessons did you learning from doing this? How can your work be improved? Did you include any improvements in your work and what impact did it have?

*1 – Presenting Opposing WordClouds*

This Text Mining assignment had originally intended to look at the difference in language tone across news websites covering the same, or similar, political events. Although the approach and subject matter of the assignment changed significantly, there were some research papers relating to Text Mining of news outlets that continued to provide guidance on the use and presentation of WordClouds.

Of particular note was an article written by in 2012 Hensinger, Flaounas and Cristianini at Bristol University entitled *The Appeal of Politics on Online Readers*[x]. The authors employed an opposing set of WordClouds to show what words most and least appealed to readers in terms of their likelihood to read an article.

This technique seemed to fit well with the objective of this assignment and I followed their approach to generate a single ‘at-a-glance’ view, from a Rotten Tomatoe movie page, of both the critic and user review WordClouds.

*2 – Bigrams and Working with Python WordCloud Parameters*

Although the initial WordClouds generated in this assignment were providing a reasonable sense of review content, the focus on displaying single words was removing some valuable context. Looking at related research on sentiment analysis from Rotten Tomatoes there was a reference in the paper from Sorostinean, M., Sana, K., Mohamed, M. and Targhi, A., 2017. *Sentiment Analysis on Movie Reviews*[x] that recommended the use of Bigrams. (A bigram is a sequence of two adjacent elements from a string of tokens.)

Researching the use of Python WordClouds, I found an article on the *towardsdatascience.com* website entitled *Generate Meaningful Words in Python[x]*. This provided some practical examples on settings for the parameters in a Python WordCloud so that I could extract meaningful short phrases from the reviews, along with single words.

The recommended setting of eliminating words with fewer than four characters and then setting the collocation threshold to ‘3’ produced more meaningful WordClouds. This is described in more detail in Section 4 of this section of the assignment.

I am conscious that I am greatly simplifying the description of the research output to which I referred, and that there was also an element of trial and error in the settings, but I felt the

Rotten Tomatoes WordCloud output was very satisfactory.

*3 – Calling APIs vs. Web Scrapping*

My initial Python code used the *BeautifulSoup* library to scrape the data directly from the Rotten Tomatoe web page for the given Marvel movie.

This has a immediate limitation in that it only returned the text for the reviews actually visible on the first web page. There were a number of online resources that provided guidance on URL manipulation as a possible solution. However, the Rotten Tomatoes website is constantly evolving in terms of its structure and the BeautifulSoup approach was not proving very robust in the face of these changes.

Looking over online research papers in the area of movie reviews and text mining, I found a others had begun their projects[x] by using the publicly available APIs that websites such as Rotten Tomatoes provide for data retrieval. Following this trail of breadcrumbs led to online technical specifications and sample code that greatly simplified the extraction of Rotten Tomatoes review data.

1. **References**

Use the IEEE Referencing style. See this guide for details. <https://libraryguides.vu.edu.au/ieeereferencing/gettingstarted>

# TASK 2 - *<insert select Task Name here e.g. Association Rules Problem>*

1. **Definition of Problem**

Clearly state the problem definition, what type of data mining task is it, where was the data set sourced from, etc.

1. **Data Exploration & Descriptive Analytics**

Include any data insights discovered

1. **Data Preparation**

Include details of any data cleaning, transformations, data enrichment, feature engineering, feature reduction, etc

1. **Details of Algorithms & Configurations**
2. **Model Performance Metrics & Evaluation of Results**
3. **Comparison with other Research**

Compare your results to at least three other researchers (maximum of five) who used the same data set. What lessons did you learning from doing this? How can your work be improved? Did you include any improvements in your work and what impact did it have?

1. **References**

Use the IEEE Referencing style. See this guide for details. <https://libraryguides.vu.edu.au/ieeereferencing/gettingstarted>

# TASK 3 - *<insert select Task Name here e.g. Time Series Analysis Problem>*

1. **Definition of Problem**

Clearly state the problem definition, what type of data mining task is it, where was the data set sourced from, etc.

1. **Data Exploration & Descriptive Analytics**

Include any data insights discovered

1. **Data Preparation**

Include details of any data cleaning, transformations, data enrichment, feature engineering, feature reduction, etc

1. **Details of Algorithms & Configurations**
2. **Model Performance Metrics & Evaluation of Results**
3. **Comparison with other Research**

Compare your results to at least three other researchers (maximum of five) who used the same data set. What lessons did you learning from doing this? How can your work be improved? Did you include any improvements in your work and what impact did it have?

1. **References**

Use the IEEE Referencing style. See this guide for details. <https://libraryguides.vu.edu.au/ieeereferencing/gettingstarted>

# TASK 4 - *<insert select Task Name here e.g. Data Ethical Issues >*

## Task 4-1 : <Title of Case Study)

1. **Overview of problem**
2. **Ethical and Legal Challenges**
3. **Challenges for Data Scientist**
4. **Reflections**
5. **References**

Use one of the commonly used References and Citation formats.

## Task 4-1 : <Title of Case Study)

1. **Overview of problem**
2. **Ethical and Legal Challenges**
3. **Challenges for Data Scientist**
4. **Reflections**
5. **References**

Use one of the commonly used References and Citation formats.