University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	**Text-as-Data (H)**			
Coursework Number	**1**			
Deadline	**Time:**	**11:59pm**	**Date:**	**14th March 2022**
% Contribution to final course mark	**20%**			
Solo or Group ✓	**Solo**	**X**	**Group**	
Anticipated Hours	**20 hours**			
Submission Instructions	**As per specification below.**			
Please Note: This Coursework cannot be Re-Assessed				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline the work will be assessed in the usual way;

(ii) An extension can be requested for a further 5 working days, taking the submission up to 10 working days after the deadline. To request this you should email both lecturers (a reply from either lecturer will be enough to confirm the extension) with an explanation of why the extension is being requested.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form via https://studentltc.dcs.gla.ac.uk/ for all coursework

Text-as-Data Coursework

Introduction

NOTE: If you are a Masters student taking Text-as-Data (H), see the Masters-specific version of this coursework.

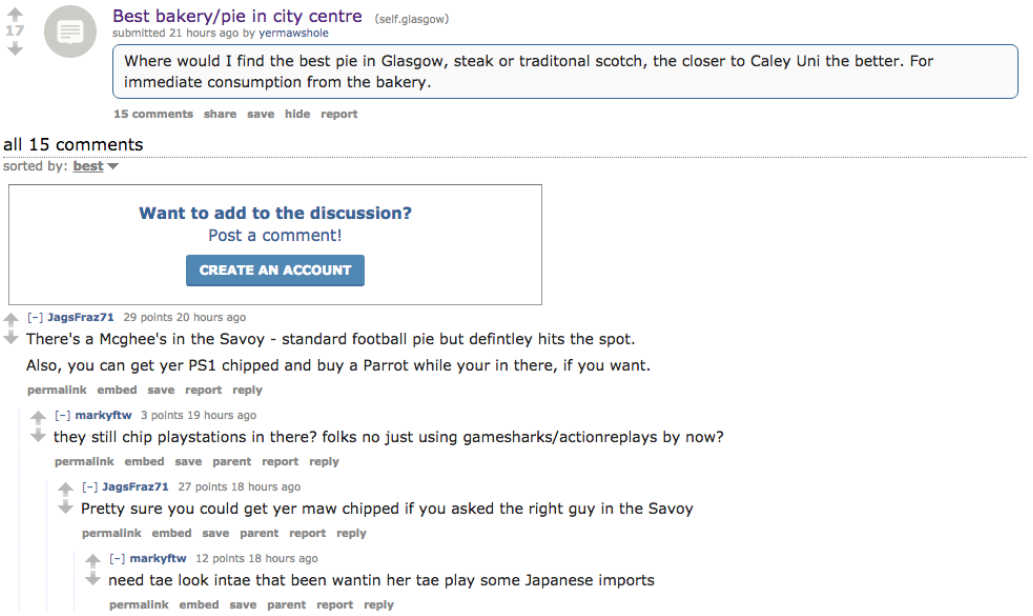
The TaD coursework aims to assess your abilities relating to techniques discussed in the course. The objective is to assess your ability in text processing techniques, and applications to text classification. In particular, this coursework is based on the Reddit datasets already explored in the lab and addresses a supervised text classification task, namely subreddit classification.

Your work will be submitted through Moodle and will comprise a **PDF report and your Jupyter/Colab notebook (as a separate .ipynb file)**. Please use the following skeleton to type your report: https://bit.ly/3gX8m9K (Also located on Moodle)

This is an **individual exercise**, and you should work independently. If you have questions concerning this document, you are encouraged to contact the course lecturers as soon as possible. There is also an FAQ document on Moodle that will be updated in response to students' questions.

Subreddit Classification [35 marks]

Your aim in this task is to predict which **"subreddit"** each post belongs to. For instance, given the thread below, your aim would be to correctly classify it into **/r/glasgow**. To do this you will train a text classification model from the training data, optimise parameters with the validation data and finally evaluate using the testing data. You will use the Reddit data used in labs. For this coursework, it has been split up into training, validation and test set splits. **Do not change this data split.**



Notebook & Dataset

We provide a skeleton Colab which contains the code to download the dataset and a broad structure for the various questions. Please use it! You may add any additional code and text cells as needed to complete the tasks and answer the questions.

Skeleton Colab Link: https://colab.research.google.com/drive/1gcI1lCx8UQzANqo7rDSCVZL-Z253nzcx?usp=sharing

The Skeleton Colab is structured into three questions with various subparts. Please structure your code under each heading. Remember to **Save a Copy to Drive** and complete your name and student number at the top.

Q1 - Comparing Classifiers [10 marks]

Use the text from the reddit posts (known as "body") to train classification models using the Scikit Learn package. The labels to predict are the *subreddit* for each post. Conduct experiments using the following combinations of classifier models and feature representations:

- Dummy Classifier with strategy="most_frequent"
- Dummy Classifier with strategy="stratified"
- LogisticRegression with One-hot vectorization
- LogisticRegression with TF-IDF vectorization **(default settings)**
- SVC Classifier with One-hot vectorization (SVM with RBF kernel, default settings))

(a) An important first step for any machine learning project is to explore the dataset. Calculate counts for the various labels and comment on the distribution of labels in the training/validation/test sets **[1 mark]**

(b) Implement the five classifiers above, train them on the training set and evaluate on the test set. Discuss the classifier performance in comparison to the others and preprocessing techniques **[6 marks]**

> For the above classifiers report the classifier accuracy as well as macro/weighted-averaged precision, recall, and F1 (to three decimal places). Show the overall results[1] obtained by the classifiers on the training and test sets in one table, and highlight the best performance. For the best performing classifier (by weighted F1 in test set) Include a bar chart graph with the F1 score for each class - (subreddits on x-axis, F1 score on Y axis).
>
> Analyse and discuss the effectiveness of the classifiers. Your discussion should include how the models perform relative to the baselines and each other. It should discuss the classifiers' behaviours with respect to:
>
> 1) Appropriate model "fit" (how well is the model fit to the training/test dataset),
>
> 2) Dataset considerations (e.g. how are labels distributed, any other dataset issues?)
>
> 3) Classifier models (and their key parameters).

(c) Choose your own classifier/tokenization/normalisations approach, and report on its performance with respect to the five previous ones on the test set. **[3 marks]**

> You should describe your selected classifier and vectorization approach including a justification for its appropriateness.

[1] Accuracy and weighted average precision / recall / F1

Q2 - Tuning and Error Analysis [10 marks]

In this task you will improve the effectiveness of the **LogisticRegression with TF-IDF vectorization** from Q1.

(a) Parameter tuning - Tune the parameters for both the vectorizer and classifier on the validation set (or using CV-fold validation on the train). **[5 marks]**

- Classifier - **Regularisation** C value (typical values might be powers of 10 (from 10^-3 to 10^5)
- Vectorizer - Parameters: **`sublinear_tf`** and **`max_features`** (vocabulary size) (in a range None to 50k)
- Select another parameter of your choice from the classifier or vectorizer

  Your search does **not** need to be exhaustive. Changing all parameters at once is expensive and slow (a full sweep is exponential in the number of parameters). Consider selecting the best parameters sequentially. The resulting tuned model should improve over the baseline TF-IDF model. Report the results in a table with the accuracy, macro/weighted-averaged precision, recall, and F1 on the **test data**. Discuss the parameters and values you tried, what helped and what did not and ***explain why*** **this may be the case.**

(b) Error analysis - Manually examine the predictions of your optimised classifier on the test set. Analyse the results for patterns and trends. Hypothesise why common classification errors are made. Report on your error analysis process and summarise your findings. **[5 marks]**

Q3 - Feature Engineering [10 marks]

In this task your goal is to add **two features** to (try to) improve *subreddit* classification performance obtained in Q2.

You must implement and describe two new classifier features and add them to the tuned model from Q2. Examples include adding other properties of the posts, leveraging embedding-based features, different vectorization approaches, etc, (**This is your chance to be creative!**). As before, report the results in terms of evaluation metrics on the test data. Additionally, include a well-labelled confusion matrix and discuss the result in reference to Q2 and what helped (or didn't) and why you think so. In summary:

(a) Propose two features of your own, along with your rationale behind your choice. **[4 marks]**

(b) Train, validate and test models that incorporate combinations of your features, and briefly report on the evaluation metrics **[2 marks].**

(c) Provide performance analysis (intrinsic evaluation metrics) of the model with your proposed features, and discuss (In your opinion), why did it work / or didn't work / what could be done to improve (Try to connect it to the lecture material) **[4 marks].**

Discuss results obtained.

Colab Report Quality [5 marks]

Quality of the report (organisation, correct spelling, presentation, use of appropriate diagrams (evaluation metrics; confusion matrices; etc): **[5 marks]**

Submission Process

You should submit your colab notebook and report separately. We provide a skeleton Colab document to get you started. Additionally, find a skeleton report to structure the report. Please keep the major headings for question numbering complete.

Skeleton Colab Link: https://bit.ly/3s0Q51r

Skeleton Report: https://bit.ly/3gX8m9K

Remember to **Save a Copy to Drive** and complete your name and student number at the top. For submission, you must submit two separate files:

- Save your Colab to a ipynb file (File -> Download -> Download ipynb)
- Your report as a PDF (File -> Print and use a PDF printer)

Make sure that the code has run and the output is visible in the .ipynb files.

Moodle Submission

The submission will be through Moodle and will include the above-mentioned report and colab notebook as separate files. Please name the files starting with your matriculation number.

The deadline to upload on Moodle is **Monday 14th March 2022 @11:59pm**.