Durham University

Computer Science

# Summative Assignment

| | |
|---|---|
| **Module code and title** | COMP42415 Text Mining and Language Analytics |
| **Academic year** | 2023/24 |
| **Submodule title** | - |
| **Submodule credits** | 15 |
| **Coursework title** | Text Mining and Language Analytics coursework |
| **Coursework % of submodule** | 100% |
| **Lecturer** | Dr Stamos Katsigiannis |
| **Deadline*** | 15/03/2024 14:00 (UK time) |
| **Hand in method** | Jupyter Server |

| | |
|---|---|
| **Additional coursework files** | *"food_reviews.zip"* |
| **Required submission items and formats** | *A Jupyter notebook with the required Python code and report, named "COMP42415_2324_coursework.ipynb". Supplementary files if needed.* |

*** This is the deadline for all submissions except where an approved extension is in place. Late submissions received within 5 working days of the deadline will be capped at 50%. Late submissions received later than 5 days after the deadline will received a mark of 0.**

# COMP42415: Text Mining and Language Analytics coursework

## Content and skills covered by the assignment

- Understand advanced concepts of Natural Language Processing (NLP).
- Have a critical appreciation of the main strengths and weaknesses of a range of NLP methods and understand how to use them.
- Have a critical appreciation of how to prepare textual datasets for analysis.
- Understand how to manipulate potentially large datasets in an efficient manner.
- Be able to write computer programs for NLP in Python using industry-standard packages.
- Be able to select appropriate data structures for modelling various NLP scenarios.
- Be able to select the appropriate algorithms for a given NLP problem.
- Be able to prepare, train, evaluate and deploy machine learning models for NLP.
- Effective written communication.
- Planning, organising and time-management.
- Problem solving and analysis.

## Requirements

Students are expected to work on the coursework **individually**.

This assignment requires you to design, explain and justify your proposed solution for a Natural Language Processing (NLP) scenario. The solution should be implemented using the Python (3.x) programming language and also requires a written report to demonstrate how and why you designed the proposed solution, as well as a thorough performance evaluation of it.

## Scenario

In this imaginary scenario, you are a data scientist for a marketing company. Your company has asked you to create machine learning models that given a written review about a food, they can predict the food's rating by online users. To succeed in your assignment, you have to use the provided food reviews' dataset and create suitable machine learning models for predicting the user rating of a food review, as described below. The deliverables for your assignment consist of a **Python implementation** for training the proposed models and using them for predictions, as well as a **written report** that justifies your decisions and provides a performance evaluation of your proposed solutions.

## Dataset

The dataset is stored in the "*food_reviews.zip*" file and consists of a comma-separated file (csv) with 540,031 food reviews from Amazon. Each review is annotated with a rating between 1 and 5. The dataset file is organised into 3 columns, as follows:

| Column | Description |
|---|---|
| Score | Rating between 1 and 5. |
| Summary | Brief summary of the food review. |
| Text | The food review's text. |

## Python implementation (70% of Total Marks)

You are asked to use the provided dataset in order to develop and evaluate machine learning models for predicting the rating of a food review (classification). Please create a single Jupyter notebook to solve all the required tasks. The required tasks are the following:

1. Prepare the dataset by applying any pre-processing or cleaning steps that you consider as necessary. Then, split the dataset into a training set containing 70% of the samples and a test set containing 30% of the samples. Follow an appropriate strategy for the split. You must use these training/test sets for all the models in this coursework. **(10%)**
2. Implement a Naïve Bayes model for predicting the rating of a food review. Train your model on the training set and test it on the test set. Use an appropriate text representation. **(5%)**
3. Implement a *k*-Nearest Neighbours model for predicting the rating of a food review. Train your model on the training set and test it on the test set. Use an appropriate text representation. You must select the best *k* by examining the performance of the model for $k \in \{1, 3, 5, 7\}$, using an appropriate cross-validation approach. Create a plot for *k* vs. classification performance to justify your choice. **(10%)**
4. Implement a Convolutional Neural Network (CNN) model for predicting the rating of a food review. The model must have at least two convolutional layers. Train your model on the training set and test it on the test set. Use an appropriate text representation. **(13%)**
5. Implement a Recurrent Neural Network (RNN) or a Long Short-Term Memory (LSTM) model for predicting the rating of a food review. The model must have at least two RNN/LSTM layers. Train your model on the training set and test it on the test set. Use an appropriate text representation. **(12%)**
6. Compute the confusion matrix, accuracy, F1-score, precision and recall for each model. **(10%)**
7. Store the **four** trained models in files and implement a function "predict_food_review(text, model)" that given a text string ("text") and model filename ("model"), it will load the pre-trained model, and predict the food review rating of the input text. The function should be able to work without requiring to rerun all or part of your code. **(10%)**

**Note:** You are strongly advised to use the Pandas library for loading and manipulating the dataset. **Your code must run successfully on the Jupyter server used for the coursework's submission**. You are strongly advised to do your code development directly on the Jupyter server or at least test your code on it before submission.

## Written Report (30% of Total Marks, 1500 words max)

You are asked to provide a written report about the developed NLP solution. The report should be divided into the following five sections: (1) Dataset, (2) Data preparation, (3) Machine learning models, (4) Experimental results, (5) Discussion. The following are required for each section, respectively:

1. Critical discussion about the dataset (suitability, problems, class balance, etc.). **(6%)**
2. Description and justification of the data preparation step(s) used. **(6%)**
3. Description and commentary on the machine learning architectures used, including a description and justification of the text representation method(s) used. **(7%)**
4. Detailed performance evaluation of the trained machine learning models in terms of the computed performance metrics. **(5%)**
5. Critical discussion on the achieved results, including potential limitations and usage instructions/suggestions. **(6%)**

**Attention:** The report should be included at the end of the Jupyter notebook using markdown cells. The report should be self-contained, i.e. write the report as if it was not bundled with your code.

## Examiners expectations

What the examiners expect from your software implementation:

- Your program must be runnable – a program that partially works or does not run at all will receive no mark.
- You are asked to use Python 3.x.
- Your source code should be documented with comments, making it to be as easily followed as possible.

What the examiners expect from the report:

- Your report needs to be professional, and the language should be scientific.
- Your report should provide justification for the design decisions you made in your solution.

## Word Limit policy

Tables and figures are excluded from the word limit. Examiners will stop reading once the word limit has been reached, and work beyond this point will not be assessed. Checks of word counts may be carried out on submitted work. Checks may take place manually and/or with the aid of the word count provided via electronic submission.

## Plagiarism and collusion

Your assignment will be put through the plagiarism detection service on Learn Ultra and the submitted Python code will be checked using a programming plagiarism detection tool.

Students suspected of plagiarism, either of published work or work from unpublished sources, including the work of other students, or of collusion will be dealt with according to the Computer Science Department and University guidelines.

Please check https://durhamuniversity.sharepoint.com/teams/LTH/SitePages/6.2.4.aspx and https://durhamuniversity.sharepoint.com/teams/LTH/SitePages/6.2.4.1.aspx for more information.

## Use of Chat-GPT or other generative AI models

You are not allowed to use Chat-GPT or other generative AI models in order to create parts of your code or your report.