QBUS6850 Group Assignment

Due Date: 17:00pm, 3 November 2023, AEST

Value: 25% of the total mark

Rationale

This assignment has been designed to help students develop valuable communication and collaboration skills when working in a team, and to allow students to practice state-of-the-art machine learning techniques and apply their machine learning skills on real-world datasets.

Instructions

1. **Required submission items via Canvas:**
    1) **ONE** written report (PDF format, through Canvas > Assignments > Report Submission (Group Assignment)).
    2) **ONE** Jupyter Notebook ".ipynb" or ".py" file (through Canvas > Assignments > Code Submission (group Assignment)).
    3) **ONE** Test results ".csv" file (through Canvas > Assignments > Challenge Results Submission (Group Assignment))
2. We suggest the group leader should submit the report and other required files on behalf of the group.
3. The late penalty for the assignment is 5% of the assigned mark per day, starting after 17:00pm on the due date. The closing date **Friday, 10 November 2023, 17:00pm** is the last date on which an assessment will be accepted for marking.
4. **Anonymous Marking**: As per anonymous marking policy, please only include Group ID and Student IDs of all group members in the submitted report. Do **NOT include names**. The file names of your submission must follow the following format: **GroupXX_QBUS6850_2023S2.** Replace "XX" with your Group ID in two digits.
5. The main text of your report should have a maximum of 12 pages, excluding cover page, duty declaration page, references and appendices (if any).
6. Your report should give full explanation and interpretation of any results you obtain. Output without explanation will receive **zero** marks.
7. Presentation of the assignment is part of the assignment. Markers will assign **10%** marks for clarity of writing and presentation.
8. Numbers with decimals should be reported to the **third-decimal point**.

Key Rules

- Carefully read the requirements for each part of the assignment and follow any further instructions (if any) announced on Canvas.
- Failure to read information and follow instructions may lead to a loss of marks.
- You must use Python for the assignment.

- Reproducibility is fundamental in machine learning, so you are required to submit your code files that can generate the results in your report. Not submitting your code would lead to **a loss of 50%** of the assignment mark.
- The University of Sydney takes plagiarism very seriously. Please be warned that plagiarism between individuals/groups is always obvious to the markers and can be easily detected by Turnitin.
- Each group will be awarded a group mark as per the marking criteria. Each group is required to record at least 3 meeting minutes. In case of a dispute in a group, the unit coordinator will request minutes of the previous meetings. Individual marks may be applied if there is a dispute and the quality or quantity of contributions made by individuals are significantly different.

Project Description

Recent years have witnessed an explosive growth of user review data generated across social media (e.g., forum discussions, blogs, Twitter) on the Web. Individuals and businesses are increasingly using such data to better understand their audience and make better decisions. Through analysing public opinions towards their products or services, businesses can develop comprehensive insights to customers' experience, and use this to improve their offerings and build a better brand and improve their business.

Suppose you are now working as a Data Science Team in a flight centre, you are tasked to practice your machine learning skills on a real-world dataset and build various models for automatically analysing travellers' sentiment towards different airlines. Your analysis will provide airlines with actionable insights that are crucial for enhancing customer satisfaction.

Datasets

The datasets provided to your team contains a collection of travellers' reviews about different airlines. The datasets are organized in two data files: `review_train.csv` and `review_challenge.csv`. Only `review_train.csv` contains the target variable "airline_sentiment" that takes three different sentiment values of "positive", "neutral" and "negative". The meanings of all features are provided in the metadata file `review_metadata.json`. You should train/validate your models using `review_train.csv` and apply your final model on `review_challenge.csv` to generate the prediction results.

You will need to undertake the following tasks and accordingly summarize your findings and analyses in your report. Please note that the tasks are deliberately designed with high flexibility for this real-world problem. This thus gives more freedom for your group to come up with your creative solutions.

Task A (15 marks)

You first need to conduct a thorough exploratory data analysis (EDA) to gain a better understanding of the given datasets and business objectives. This includes but not limited to: checking/dealing with missing data, visualizing the distributions of features, identifying relevant features that can better distinguish different target values, and conducting feature correlation analysis, etc. Carefully present your analysis and findings in your report.

Task B (15 marks)

As your first attempt, you want to apply conventional vector-based machine learning algorithms and assess the feasibility of the prediction task. For this task, you are required to build your models based on "review_text" only.

Given that reviews are represented as raw text, you need to pre-process raw text and then perform feature engineering before building your models. Decide on what engineered text features (e.g., BoW, TFIDF) and/or your own engineered features to train your chosen model of either Random Forest or Gradient Boosting. Justify your choice of feature engineering strategies and tune appropriate hyperparameters that apply for your chosen model. You need to select appropriate evaluation metrics and model evaluation strategies to validate your model. Document your analysis and discuss your findings.

Task C (20 marks)

For this task, you would like to build a supervised deep learning model for fake news detection. You are required to build a vanilla recurrent neural network (RNN) (i.e., consisting of a single RNN layer) based on "review_text" only. You may follow the same pre-processing procedure in Task B and make sure the resulting data are suitable for building your RNN model. It is your choice to set parameters (e.g., maximum sequence length) with data-based evidence.

Carefully design your network architecture and specify detailed configurations of all layers. Tune at least three hyperparameters that you find most affect your model performance. Document your attempts and report on the hyperparameter settings. Properly evaluate your model performance and compare with the model you have built in Task B.

Task D (10 marks)

Based on the models you have built from Task C, you are required to explore the influence of using pre-trained word embeddings on model performance. For this task, you can use GloVe (Global Vectors for Word Representation) pre-trained word vectors and the detailed information can be found at **https://nlp.stanford.edu/projects/glove/**. You can download the glove.6B version from Canvas: `glove.6B.300d.zip`, where each word is represented by a 300-dimensional embedding vector.

You are required to write your own program that incorporates pre-trained word vectors into your model training. Explain your implementation and report model performance as compared to the models from Task C. Carefully report your findings and analysis.

Task E (25 marks)

Based on your attempts, you are now required to make further improvements over the models you have built so far. You may consider two pathways:

- **Feature Engineering.** You might incorporate additional features with appropriate feature engineering to rebuild your models. Your choice should be justified based on the evidence from the data.

- **Build new models**. You might also change your network architecture for incorporating additional features or build more advanced models. Your choice of decisions should be properly justified.

When building your new models, you must properly validate your model performance and tune appropriate hyperparameters that apply. Simply building a model without any consideration of validation and hyperparameter tuning does not meet the minimum requirements.

You should demonstrate evidence of your efforts and you will be assessed based on the depth of your exploration. Provide insights and analyses about what has worked and what has not. Report on your improved models and make appropriate comparisons with the previous models built from Tasks B, C and D.

Task F (5 marks)

Finally, according to your analysis, decide on your best model and apply it to the challenge dataset `review_challenge.csv`. You are asked to report the prediction results on the challenge dataset provided, which will be evaluated against the ground-truth labels. Save your prediction results into a csv file containing two columns, where the first column "review_id" indicates the review IDs from `review_challenge.csv` and the second column "airline_sentiment" indicates the predicted sentiment values ("positive", "neutral", or "negative"). Name your file as **GroupXX_QBUS6850_2023S2.csv**. An example file of challenge results `challenge_results_example.csv` is also provided on Canvas. Note that, the number of rows in your csv file should be the same as in `review_challenge.csv`. As we will run programs to evaluate your final model performance, submissions with incorrect file names and formats will lead to zero marks for this part.

The prediction results on the test data will be assessed to decide your performance among the entire class (competition!). **F1-score** will be used as the test score for competition.

Competition: A competition will be run among the entire class to rank the performance of your models on the test data provided. The top 3 groups will be awarded with **bonus marks** to top up their overall assignment mark: the first place will receive an extra 6 marks, the second place an extra 3 marks, and the third place an extra 1 mark.

Presentation (Report and Code) (10 marks)

The overall project presentation includes:

1. Your runnable Python code for all the tasks. Make your code as concise as possible and add comments when necessary to explain the functionality of your code segments.
2. Your final report consisting of:
    a) A cover page with a list of members Student IDs [don't add your names] (this is not counted towards the overall page limit)
    b) A page with member duty declaration showing the details of tasks conducted by each member [use Student ID only for each member] (this is not counted towards the overall page limit).
    c) The report main body: The main body should be in the form of a technical report within 12 pages **excluding** references, with font size no smaller than 11pt. Think about the best and most structured way to present your work, summarise the

procedures implemented, and explain your results/findings. Keep in mind that making good of your audience's time is an essential skill. Make sure your writing is concise and clearly conveys your points.

d) Any relevant figures and tables should be clearly and appropriately presented.
e) The reference should be in APA 6th or APA 7th format. Details of referencing styles can be found at the library's website:
https://libguides.library.usyd.edu.au/c.php?g=508212&p=3476063
f) Appendices. Those supplementary information, extra figures/tables etc that you believe are necessary. Appendices are not counted towards the overall page limit.

The following resources would be helpful for you to write a technical report:

- **https://students.unimelb.edu.au/academic-skills/explore-our-resources/report-writing/technical-report-writing**.
- **https://www.monash.edu/rlo/assignment-samples/engineering/eng-writing-technical-reports**