IS41070 Project

IS41070 Machine Learning Foundations
Academic Year 2023/2024

Deadline: Wednesday 17th July

Overview:

The objective of this project is to perform a classification task using machine learning. The classification task consists of classifying news articles into the correct topical category. Students are expected to use the concepts learned in class, but can also go beyond what has been taught in the module via external research.

Each student must submit **their own work**. Any code used from the web should be referenced. Students are also encouraged to **ask questions about the assignment on Brightspace** and to help each other.

The project is divided into three parts: (i) data understanding, (ii) data preparation & modelling and, (iii) evaluation.

NB - The assignment should be implemented as Jupyter Notebooks (format .ipynb) and not as script files (format .py). Your notebooks should be clearly documented, using comments, explanatory variable names, and Markdown cells to explain the code and interpret the results of your analysis. **Explanatory material is a key portion of the assessment -** you do not need to comment every single line, and your explanations may be brief, but the **purpose and operation** of your code should be clear. Any **complex code section** should receive its own comment - if you are in doubt about whether a code snippet should receive a comment, **err on the side of caution and comment it.** It will not add much to your workload, and will make your code much more understandable - remember, the person reviewing your code will have **no familiarity** with your work**.** The purpose of your code (e.g. a block of code that cleans a dataframe) should always be documented, preferably in a preceding markdown section.

Please see the instructions for submitting your work at the end of file.

Task 1. Data understanding (20 points)

1. Download a csv file with your personal dataset on this link: https://www.dropbox.com/scl/fo/wxjdvg30q6d2gv1mv2gqj/AD2-wtDRiWzawV4BM3ORgrg?rlkey=iy1hs8c79np2t8q41n0u2zbwe&st=9azf6rm2&dl=0

Use the excel sheet uploaded with the assignment as a guide. The sheet has your name and the filename you have been assigned so download the file with name in front of your name.

You received a small portion of a larger dataset that was collected by Rishabh Misra. The original dataset is available here. The dataset has many categories, but your dataset will contain **only two categories.**

2. Load this dataset.
3. Perform an exploration of the data.
    i) Perform an analysis of the most common terms for each category. Consider whether you would like to preprocess the text before analysing the most common words used.
    ii) Analyse other features in the dataset and their relationship with the label you are trying to classify (the category of the article). Analyse the length of the sentences in each category.
    iii) Check the dataset for blank values, incorrect data, and outliers.
    iv) Comment on your observations.

Task 2. Data Preparation & Modelling (20 points)

4. Splitting the dataset into training, development and test sets.
    i) Choose an appropriate split for your data. Make sure you have split the dataset into training, validation and test sets. Comment on your choices. The test set should not be used for parameter tuning and should only be used on the last item of Task 3.
    ii) Save your data as separate csv files (train.csv, valid.csv and test.csv).

5. Load your train.csv and valid.csv files. Apply appropriate preprocessing steps to create a numeric representation of the documents, suitable for classification. Consider whether you would like to apply any transformation to the text such as lowercasing, removing punctuation, stemming, etc. Consider whether any other feature (other than the review itself) needs any preprocessing. **Always explain the reason behind your choices.**
6. Build binary classification models using two classifiers covered in our lectures to distinguish between your two news categories. Comment on your choices for the classifier and parameters used in each classifier.
7. Build or apply an end-to-end classifier using deep learning. For this task, you can either train your own deep learning model from scratch or fine-tune an existing model.

Task 3. Evaluation (50 points)

8. Choose a primary metric that will be used to evaluate your models. Justify your choice. Comment on what is a good benchmark for this task.
9. Evaluate the performance of each model developed on Task 2 (items 6 and 7) on your train and validation sets. How does the performance on the train set compare to the validation set? Comment on the performance of the classifiers/models.

10. Perform an error analysis for each model tested on the previous item. Comment on your results. Consider things like: did the different models classify the same sentences incorrectly? What have you learned from this analysis?
11. Apply at least one change to each classifier/model developed on Task 2 and redo your evaluation. Think of a change that can help improve the metric you are using to evaluate your models. This change can either be a change of a parameter, or a different preprocessing or any other change you may find interesting to implement after doing an error analysis. Depending on your primary metric, you may want to consider strategies to address the imbalance in your dataset. Save these models in an appropriate format. Comment on your choices and results. Could you achieve the benchmark you expected for this task?
12. Merge your train and validation sets and perform **cross validation** using the classifiers from item 11. Comment on your results.
13. Choose the best model from the previous item, load it using the files created on item 11 and apply it to the test set (test.csv). Make sure you are preprocessing the test set exactly the same way you preprocessed the data you used to train the model. Are the results in your metric of choice similar to the one you obtained for the validation set? Comment on your results.
14. Retrain the best model from Task 3 (item 11) using the train and validation datasets and now apply to the test set. Did training the model with more data make any difference? Comment on your results.
15. Your code should be perfectly reproducible. Make sure you are getting the same results when you run it for any model that is deterministic. Make sure you are including a requirements.txt file (a list of any additional packages used).
16. Save your notebooks with all output as a html file by clicking “File” < “Download as” < ‘HTML’.

Guidelines:

- Make sure you **have cleared all output** before submitting your Jupyter notebook(s). This can be done by clicking “Cell” > “All output” > “Clear”. Your models will not be re-trained during correction, they will only be loaded so make sure you save your models and that they are loading correctly! Also, make sure your Jupyter Notebook(s) are running without errors from start to finish.
- Make sure **you don’t mention your name** anywhere in your submission.
- Submit your assignment via the module’s Brightspace page. Your submission should be in the form of a **single ZIP file** containing the data (train, valid and test sets), models and notebook(s) (i.e. IPYNB files). **Please use .zip format not .rar.**
- The assignment should be completed individually. Any evidence of plagiarism will result in a 0 grade.
- The grade awarded will depend on the complexity of the analysis and level of detail, i.e., data preprocessing, classifier evaluation and comparison etc.

- Hard deadline: Submit by end of **Wednesay 17th July 2024**
    - 1-5 days late: 1 grade point deduction, e.g. B to B-
    - 6-10 days late: 2 grade point deduction, e.g. B to C+
    - Late without notifying me a week in advance: 1 grade point deduction, e.g. B to B-
    - Assignments will not be accepted after 10 working days without Extenuating Circumstances formally approved by UCD.