CM50266 Applied Data Science

Lab 2: Sentiment Analysis of Reviews

Deadline

Lab Due	8pm 11th December 2020. (End of week 11)
Peer Assessment Due	8pm 8th January 2020. (End of week 15)
Marks	/10 (10% of overall unit mark.)

For this assignment you are required to apply skills and knowledge that you have been developing across several of the units that you have been taking so far. You are also expected to engage in further investigation and research of the topics that are covered in this assignment in order to develop your knowledge further.

Data

You are provided with a large dataset of reviews for Ford motor vehicles (car_reviews.csv) on Moodle. Each review is labelled with either 'Pos' or 'Neg' to indicate whether the review has been assessed as positive or negative in the sentiment it expresses. You should treat these labels as a reliable indicator of sentiment. You can assume that there are no neutral reviews. There are 1,382 reviews in the CSV file in total, 691 of which are positive and 691 of which are negative.

Task 1 (6 marks)

In a Jupyter notebook, implement a **Naïve Bayes** classifier using 80% (1106) of the reviews as training data. The training data should be selected at random from the full dataset. Test your classifier using the remaining 20% (276) of the reviews and report the classifier's performance using a confusion matrix.

It is important that you avoid issues of data leakage[1], meaning that your classifier should only be trained using data that it has access to from within the training data set. If there are words that only appear in the test data they should not be part of the classifier. You will need to make sure that your code is able to deal with encountering words in the test data that the classifier has not seen in the training data. It is up to you to decide how you will handle this.

Your code will need to read the review data CSV file provided. For this you are free to use either the CSV reader that you implemented for Lab 1, or any other CSV reading library that you wish (e.g. Pandas).

You will need to perform some clean up of the data before using it in your classifier. This should include:

- Identifying and excluding all punctuation and words that are not likely to affect sentiment (e.g. stopwords[2]). As an example, Natural Language Toolkit (NLTK) in Python has lists of common stopwords that you may wish to use, but you are also free to find and use other libraries or tools for this.
- ensuring that remaining words are **not** case sensitive (i.e. the classifier should not distinguish upper/lower case characters).

Your sentiment classifier should use a bag of words technique[3], in which you build a vocabulary of individual words that appear in the dataset once it has been cleaned up.

You should attempt to treat minor variations of a word (e.g. 'fault', 'faults' and 'faulty') as instances of the same word (e.g. 'fault') when you are using them in your classifier. You should investigate and implement *stemming* as a way of doing this.

---

[1] https://en.wikipedia.org/wiki/Leakage_(machine_learning)
[2] https://en.wikipedia.org/wiki/Stop_word
[3] https://en.wikipedia.org/wiki/Bag-of-words_model

For each review you should create a vector as input for your classifier, containing EITHER binary values indicating whether a word/stem occurs in the review OR a numerical count of the number of times each word/stem appears. As described above, vectors that are used to train the classifier should only include words that appear in the training data (and not words that only exist within the test data).

Note: You do not need to code everything required from scratch. For this lab exercise you are encouraged to make use of existing libraries for all parts of the tasks. For example, you may find the MultinomialNB classifier in scikit.learn and natural language processing tools such as NLTK and spaCy useful for this task.

It is also important to note that there is no single correct answer in terms of the output and performance of your classifier. This will depend on the choices you make about how you deal with the data at each stage of the process – your markers will not be looking for a specific level of performance, rather that you have taken appropriate steps and implemented them correctly.

Your solutions to Task 1 will be assessed according to the following criteria:

Assessment Criteria for Task 1 **A mark should be awarded for each of the following criteria that are satisfied. Marks should not be awarded for a particular criterion if there are errors identified in the code for that particular aspect of the software.**	Marks Awarded
Does the code produce some output (e.g. using print function) to clearly demonstrate that words and punctuation, which are unlikely to affect sentiment, have been excluded from the sentiment classifier AND that the remaining words are not being handled in a case sensitive way?	1
Does the code produce some output, which clearly demonstrates that words with the same stem have been appropriately recognised and treated as variations of the stem? This should be demonstrated for at least 3 different stems.	1
Does the code produce some output to demonstrate that a vector has been created for each review, where each element in the vector represents EITHER a binary variable indicating the presence of a word/stem in a review OR the number of times that a word (or word stem) appears? Note that the output does not need to show the vector for all reviews, this only needs to contain a small sample of reviews.	1
Does the code clearly show that an appropriate Naïve Bayes model has been used for classification, either through the use of an existing library or coded from scratch.	1
Does the code clearly show that 80% of the data has been used to train the classification model, and that the remaining 20% of the data set has been used as test data? AND does it show that only the training data has been used up to the point where the model has been trained? AND is the code able to cope with words that appear in the test data but not in the training dataset?	1
Does the code output a confusion matrix demonstrating the performance of the Naïve Bayes classifier? The confusion matrix must clearly indicate the proportion of True Negatives, False Positives, False Negatives and True Positives.	1

Task 2 (4 marks)

Identify and research a way to improve on your solution to Task 1, that you would expect to do better at classifying the sentiment of the reviews.

You may either:

- identify an alternative classification algorithm, or
- apply modifications to the Naïve Bayes implementation, for example trying different classification of different size n-grams (multi-word phrases). Implement this improvement and compare the results to your initial Naïve Bayes classifier.

Note that it is does not matter if your approach for Task 2 does not actually outperform the approach you have taken in Task 1. You will not lose marks if the performance does not improve, however you must consider and explain why this might be the case when you are addressing the final criterion below.

Your solutions will be assessed according to the following criteria:

Assessment Criteria for Task 2	Marks Awarded
Does the Jupyter notebook include a markdown/comment section that clearly explains how the approach taken in Task 2 is **expected** to improve on the solution to Task 1. Are the reasons for the expected improvements clearly justified and explained with a references (e.g. to a published source scientific paper, article, book) ?	1
Does the code include comments that clearly explain the steps that have been taken to implement the improved approach? These should be written in a way that one of your peers who may not have researched the same approach could understand.	1
Does the code implement the described approach appropriately? (i.e. does the code actually do what is described?)	1
Does the code output a new classification matrix for the "improved" Task 2 approach AND Is there markdown or comment that clearly discusses and compares the performance of the Task 1 and Task 2 classification approaches and explains whether or not the expected improvements were achieved (and why this may be the case).	1

Submission

Method: Via Moodle

Submit: Jupyter notebook with code and markdown/comments for the tasks above. Your name or user ID should NOT appear in the filename or within the code. Your code will be run as part of the peer assessment. It is in your interests to include comments or markdown in your code to assist with this. You should make sure that your code provides output/comments to demonstrate to your peer marker how you have met the marking criteria described above.

Peer Assessment Submission:

Via online form.
Complete one entry for each submission you are reviewing.
You are advised to also submit an entry for your own work.

Peer Assessment

This unit will make use of peer assessment. This means that after the initial deadline for a piece of coursework you will be allocated the work of three other students to examine and assign a mark. This will allow you to see how others have tackled the same problem. The purpose of this is to expose you to issues you may not have identified for yourself and improve your understanding of the problem being tackled.

You will be provided details of how to download the three submissions. You are expected to examine these and compare them to the assessment specification given in this document. Each of the criteria is designed to be a simple pass/fail assessment where the submission either meets the requirement or it does not. Where any criteria are not met, you must indicate why you have reached this conclusion.

You will be given a link to an online form where you can submit an entry for each submission you examine. You should also submit an entry for your own work. You are strongly recommended to assess your own work after you have reviewed the work of the other students. You must submit all the forms by the peer assessment deadline.

There are no additional marks for completing the peer assessment. However, a penalty of up to 50% will be applied to your lab mark should you fail to complete the peer assessment satisfactorily.

A satisfactory assessment entry means you will have completed a form for each submission allocated to you and provided a valid justification for each of the criteria you have labelled as not met.

The work you submit should be anonymous and not include your name or userid. You should remove any reference to your username in any pathname in your code. Replace it with 'username'. You must not engage in discussion of your mark or the marks you will allocate to your peers with your peers. You should report any attempt by others to influence the marking process.

Mark Calculation

Your mark will be calculated in the following way:

1. The two closest peer marks given will be used. If the three marks are equally spaced, the pair closest to your own estimate will be used.

2. If your own mark estimate lies above the peer marks you will receive the mean of the peer marks.

3. Otherwise, you will receive the mean of the two highest marks. (The two peer marks and you own estimate.)

Your mark will be returned to you once this processing has been done. You will also receive the details of the marks allocated by your peers. This will include their reasoning. This is a provisional mark. If you do not consider the mark to be fair, you can contact the lecturer and ask for it to be reviewed. Your work will be re-marked and where the lecturer determines a different mark, the peer marking will be checked and any unsatisfactory marking will have the penalty applied. Should your request for a review not be justified, a penalty may be applied to your mark as you will have further demonstrated that you have not properly understood the material or the feedback you have received.

After the review period the coursework mark will be finalised. To maximise your marks, you should attempt to be as accurate in the marking of both the peer work and your own.

Extensions

If any student is granted an extension, they will still have to undertake peer marking of others work after their updated deadline, with appropriate extensions. Their own work may be peer marked or assessed by the lecturers/tutors depending on the availability of peer markers at that time.

KMC/SJ 2020