School of Computing and Information Systems
The University of Melbourne
COMP30027, Machine Learning, Semester 1, 2022

Assignment 2: Sentiment Classification of Tweets

Release:	Thursday 14 April 2022
Due:	*Stage I*: Friday, 13 May 2022 at 7 PM *Stage II*: Wednesday, 18 May 2022 at 7 PM
Marks:	The Project will be marked out of 20 and will contribute 20% of your total mark.
Groups:	You may choose to form a group of 1 or 2.
Main Contact:	Hasti Samadi (hasti.samadi@unimelb.edu.au)

1 Overview

The goal of this project is to build and critically analyse supervised Machine Learning methods, to predict the sentiment of Tweets. That is, given a tweet, your model(s) will produce a prediction of the sentiment that is present in the tweet. You will be provided with a data set of tweets that have been annotated with positive, negative, and neutral sentiments. The assessment provides you with an opportunity to reflect on concepts in machine learning in the context of an open-ended research problem, and to strengthen your skills in data analysis and problem-solving.

The goal of this assignment is to critically assess the effectiveness of various Machine Learning classification algorithms on the problem of determining a tweet's sentiment and to express the knowledge that you have gained in a technical report. The technical side of this project will involve applying appropriate machine learning algorithms to the data to solve the task.

The focus of the project will be the report, formatted as a short research paper. In the report, you will demonstrate the knowledge that you have gained, in a manner that is accessible to a reasonably informed reader.

2 Deliverables

Stage I:

1. **Report**: An anonymous written report, of 1900 (±10%) words (for a group of one person) or 2500 (±10%) words (for a group of two people) including reference list, figure captions and tables. Your name and student ID should not appear anywhere in the report, including the metadata (filename, etc.). *Submitted as a single PDF file through Canvas/Turnitin.*
2. **Output**: Sentiment predictions for the test instance dataset. *Submitted as a single CSV file through Canvas/Turnitin.* (You also need to submit your prediction file to the Kaggle[1] in-class competition described in section 6.)
3. **Code**: One or more programs, written in Python, including all the code necessary to reproduce the results in your report (model implementation, label prediction, and evaluation). Your code should be executable and have enough comments to make it understandable. *Submitted as a zip file through Canvas/Turnitin.*

Stage II:

1. Reviews of two reports written by other students of 200-300 words each (for a group of one person) or

[1] https://www.kaggle.com/

300-400 words each (for a group of two people).

NOTE1: Stage I submissions will be open one week before the due date. Stage II submissions will be open as soon as the reports are available (24 hours following the Stage I submission deadline).

NOTE2: If you decided to operate in groups of two ONLY one of you need to register your group via the provided link. Also all submissions (on Canvas and Kaggle) should be done by ONLY one member of the group.

3 Data Set

You are provided with a labelled training set of Tweets, and an unlabelled test set which will be used for final evaluation in the Kaggle in-class competition. In the train set, each row in the data file contains a tweet ID, the tweet text and the sentiment for that tweet[2]. For example,

Tweet_ID, "if i didnt have you i'd never see the sun. #mtvstars lady gaga", positive

The test dataset has a similar format except the rows do not include a sentiment (label). You are expected to treat each row of the dataset as an instance. For processing these instances, you need to change them to feature vectors. There are many methods for vectorizing textual instances. We have provided you with two examples.

1. **BoW (Bag of Words)**

   In the given *feature_analysis.ipynb* file, you are provided with a basic piece of code that uses the *CountVectorizer* to transform the train tweets into vectors of *Term_IDs* and their *count*. For example, with the use of *CountVectorizer* the above tweet, will be transformed into the following vector:

   [(51027, 1), (44650, 1), (40410, 1), (43384, 1), (22275, 1), (13438, 1), (20604, 1), …]

   Where 51027 is the Term_ID for the word 'you', 44650 is the Term_ID for the word 'the' and so on. You can use and edit this basic code to vectorise your *test* and *train* datasets. There are many modifications you can use to experiment with different hypotheses you may have. For example, how 'removing very frequent and/or very infrequent words' can affect the behaviour of your Machine Learning models. There are many more examples.

2. **TFIDF**

   You are also provided with a basic piece of code that uses *TfidfVectorizer* to transform the tweets as a vector of values that measure their importance using the following formula:

   $$w_{d,t} = f_{d,t} \times \log \frac{N}{f_t}$$

   Where $f_{d,t}$ is the frequency of term t in document d, f_t in the number of documents containing t, and N is the total number of documents in the collection. You can learn more about TFIDF in (Schutze, 2008). Using TFIDF the above example tweet will be transformed to the following vector:

   [(51027, 0.17*), (44650, 0.09), (40410, 0.23), (43384, 0.29), (22275, 0.22), (13438, 0.46), …]*

   Similar to the Bag of Words method, you can use and edit this basic code to vectorise your *test* and *train* datasets. Like above, there are many modifications you can use to experiment with different hypotheses you may have about how changing these features can change the behaviour of your Machine Learning models.

There are many other text vectorization methods that you can use (e.g. word2vec, Bert, etc.). You are welcome and encouraged to use as many vectorization methods as you choose. But please keep in mind that we are more

[2] Undoubtedly, there will be some tweets where you might think that the sentiment should be labelled differently, but such is the nature of Machine Learning! ☺

interested in the depth of analysis and quality of interpretation in your report, NOT the variety or complexity of the methods you have used.

4 Task

Stage I

4.1. Feature Engineering

The process of engineering or selecting features that are useful for discriminating among your target class set is inherently poorly defined. Most machine learning assumes that the attributes are simply given, with no indication from where they came. The question as to which features are the best ones to use is ultimately an empirical one: just use the set that allows you to correctly classify the data.

In practice, the researcher uses their knowledge about the problem to select and construct "good" features. What aspects of a tweet itself might indicate a tweet's sentiment? You can find ideas in published papers, e.g., (Go, 2009).

It is optional for you to use the features generated by the given code (as they are), modify the code and generate a new set of features, use other vectorizing methods to develop some different attributes or select features from the ones we generated for you.

Whatever method you choose, you have to use your features to train some models and run a few experiments on the given test data.

4.2. Machine Learning

Various machine learning techniques have been (or will be) discussed in this subject (0R, Naive Bayes, Decision Trees, k-NN, SVM, Logistic Regression, etc.); many more exist. You may use any machine learning method you consider suitable for this problem. *You are strongly encouraged to make use of machine learning software and/or existing libraries (such as sklearn) in your attempts at this project.*

In this stage your task has two phases:

- *The training-evaluation phase:* The holdout or cross-validation approaches can be applied to the training data provided. Check section 4.4 for the minimal expectations in this phase.
- *The test phase:* the trained classifiers will be evaluated on the unlabelled test data. The predicted labels of test cases should be submitted as part of the Stage I deliverable.

As a result of training different models and running a few experiments, you are expected to develop some knowledge of why you are reaching the results you do and some hypotheses of how you can change these results. For example, *removing the 'stop-words' will* reduce the noise and increase the performance; or *k-NN is working the way it does*, has something to do with the structure of the instances in this dataset; and many more.

You should then test these hypotheses with more experiments. When explaining your results, you are expected to use examples from the dataset as well as theories and findings from the lectures and published literature. You are also expected to use appropriate visualization tools (e.g., tables or diagrams) to communicate your findings professionally and academically.

4.3. Report

Your main submission for this assignment is your report. The report should follow the structure of a short research paper, as will be discussed in the guest lecture on Academic Writing. It should describe your approach and observations, both in engineering features, and the machine learning algorithms you tried. Its main aim is to provide the reader with knowledge about the problem, in particular critical analysis of your results and discoveries. The internal structure of well-known classifiers (discussed in the subject) should be mentioned if it is important for connecting the theory to your practical observations.

The following is the expected structure of the report for this assignment.

- *Introduction*: a short description of the problem and data set
- *Method:* Introduce the used feature(s), and the rationale behind including them. Explain the classifiers and evaluation method(s) and metric(s) you have used (and why you have used them). *This should be at a conceptual level; a detailed description of the code is not appropriate for the report. The description should be similar to what you would see in a machine learning conference paper.*
- *Results*: Present the results, in terms of evaluation metric(s) and, ideally, illustrative examples and diagrams.
- *Discussion / Critical Analysis*: Contextualise the systems' behaviour, based on the understanding of the subject materials (*This is the most important part of the task in this assignment*).

  Contextualise implies that we are more interested in seeing evidence of you have thought about the task and determining reasons for the relative performance of different methods, rather than the raw scores of the different methods you selected. This is not to say that you should ignore the relative performance of different runs over the data, but rather that you should think beyond simple numbers to the reasons that underlie them.
- *Conclusion*: Demonstrate your identified knowledge about the problem.
- A *bibliography*, which includes (Rosenthal, 2017), as well as references to any other related work you used in your project. You are encouraged to use the APA 7 citation style but may use different styles as long as you are consistent throughout your report.

We will provide LATEX and RTF style files that we would prefer that you use in writing the report. Reports are to be submitted in the form of a single PDF file. If a report is submitted in any format other than PDF, we reserve the right to return the report with a mark of 0.

Your name and student ID should not appear anywhere in the report, including any metadata (filename, etc.). If we find any such information, we reserve the right to return the report with a mark of 0.

4.4. Number of the models

You have the option of participating as an individual, or in a group of two. In the case that you opt to participate *individually*, you will be required to implement one baseline and at least 2 and up to 4 distinct Machine Learning models. *Groups* of two will be required to implement one baseline and at least 3 and up to 5 distinct Machine Learning models, of which one is to be an ensemble model – stacking based on the other models.

Stage II

In this stage, you will read two submissions by other students. This is to help you contemplate some other ways of approaching the project and to ensure that students get some extra feedback. For each report, you should aim to write 200-400 words total (200-300 if you work alone, or 300-400 words if you work in a group of two people), responding to three "questions":

- Briefly summarise what the author has done
- Indicate what you think that the author has done well, and why
- Indicate what you think could have been improved, and why

Please be courteous and professional in the reviewing process. A brief guideline for reviewers published by IEEE can be found here[3].

[3] https://www.ieee.org/content/dam/ieee-org/ieee/web/org/members/students/reviewer_guidelines_final.pdf

5 Assessment Criteria

The Project will be marked out of 20 and is worth 20% of your overall mark for the subject. The mark breakdown will be:

Report	16 marks
Performance of classifier	2 marks
Reviews	2 marks
TOTAL	20 marks

The report will be marked according to the rubric as published via the Canvas\Assignment2 page.

The performance of the classifier (1 mark) is for submitting (at least) one set of model predictions to the Kaggle competition; and (1 mark) to get a reasonable accuracy, e.g., better than our threshold.

You have to submit your code that supports the results presented in your report. If you do not submit an executable code that supports your findings, you will receive **zero** marks for the report section.

Since all the documents exist on the World Wide Web, it is inconvenient but possible to "cheat" and identify some of the class labels from the test data using non-machine learning methods. If there is any evidence of this, the performance of the classifier will be ignored, and you will instead receive a mark of **zero** for this component.

6 Using Kaggle

To give you the possibility of evaluating your models, even more, we will be setting up a Kaggle In-Class competition. You can submit results on the test set there and get immediate feedback on the performance of your system. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating online.

The Kaggle in-class competition URL will be announced on Canvas shortly. To participate in the competition:

- Each student should create a Kaggle account (unless they have one already) using your Student-ID
- You may make up to 8 submissions per day. An example submission file can be found on the Kaggle site.
- Submissions will be evaluated by Kaggle for accuracy, against just 40% of the test data, forming the public leaderboard.
- Before the competition close, you may select a final submission out of the ones submitted previously – by default the submission with the highest public leaderboard score is selected by Kaggle.
- After the competition close, the public 40% test scores will be replaced with the private leaderboard 100% test scores.

REMINDER: We are more interested in your critical analysis of methods and results than the raw performance of your models.

7 Assignment Policies

7.1 Terms of use

The data has been adapted from the data provided by the 2017 SemEval conference[4] under the provision that any resulting work should cite this resource:

[4] http://alt.qcri.org/semeval2017/

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). SemEval-2017 Task 4: sentiment analysis in Twitter. In Proceedings of the 11th International Workshop on semantic evaluation (SemEval '17). Vancouver, Canada.

This reference must be cited in the bibliography. We reserve the right to mark any submission lacking this reference with a 0, due to the violation of the Terms of Use.

Please note that the dataset is a sample of actual data posted to the World Wide Web. As such, it may contain information that is in poor taste, or that could be considered offensive. We would ask you, as much as possible, to look beyond this to the task at hand. For example, it is generally not necessary to read individual records.

The opinions expressed within the data are those of the anonymised authors, and in no way express the official views of the University of Melbourne or any of its employees; using the data in an educative capacity does not constitute an endorsement of the content contained therein.

If you object to these terms, please contact us (hasti.samadi@unimelb.edu.au) or (ni.ding@unimelb.edu.au) as soon as possible.

7.2 Changes/Updates to the Assignment Specifications

We will use Canvas to advertise any (hopefully small-scale) changes or clarifications in the assignment specifications. Any addendums made to the assignment specifications via Canvas will supersede information contained in this version of the specifications.

7.3 Late Submissions

Late submissions in stage I will bring disruption to the reviewing process. There will be no extensions granted, and no late submissions allowed to ensure a smooth run of the Stage II process. Submission will close at 7 pm on May 13th.

You are strongly encouraged to submit by the date and time specified above. For students who are demonstrably unable to submit a full solution in time, we may offer a solution, but note that you may be unable to benefit from the peer review process in that case. A solution will be sought on a case-by-case basis. Please email Hasti (hasti.samadi@unimelb.edu.au) with documentation of the reasons for the delay.

Any late submission of the reviews will incur a 50% penalty (i.e., 1 of the 2 marks), and will not be accepted more than 2 days after the Stage II deadline.

7.4 Academic Honesty

While it is acceptable to discuss the assignment with others in general terms, excessive collaboration with students outside of your group is considered cheating. Your submissions will be examined for originality and will invoke the University's Academic Misconduct policy[5] where either inappropriate levels of collaboration or plagiarism are deemed to have taken place.

We highly recommend (re)taking the academic honesty training module in this subject's Canvas. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy where inappropriate levels of collusion or plagiarism are deemed to have taken place.

References

Go, A. (2009). Sentiment Classification using Distant Supervision. CS224N project report, Stanford.

Rosenthal, S. N. (2017). SemEval-2017 Task4: Sentiment Analysis in Twitter. Proceedings of the 11th International Workshop on Semantic Evaluation . Vancouver, Canada: SemEval '17.

[5] https://academicintegrity.unimelb.edu.au/home

Schutze, H. M. (2008). Introduction to information retrieval. Cambridge University Press Cambridge.