School of Computing and Information Systems
The University of Melbourne
COMP90051 Statistical Machine Learning, Semester 2 2019

# Project 1: Who Tweeted That? 

| | |
|---|---|
| **Due:** | 17:00 Thurs 12th September 2019 AEST for Kaggle, Final Report, Code archive. |
| **Submission materials:** | Predictions submitted through Kaggle<br>PDF final report submitted through LMS<br>(Not marked[1]:) Group agreement, Kaggle team registration,<br>Code archive, Post-project survey through LMS |
| **Assessment criteria:** | Kaggle prediction accuracy; Kaggle team rank; Report quality |
| **Marks:** | The Project will contribute 25% of your overall mark for the subject |
| **Group submission:** | Groups of 3 students required |

## Overview

The goal of the Project is to develop skills in researching and applying methods from subareas of Statistical Machine Learning (SML) that may be unfamiliar, by leveraging the fundamental knowledge built in lectures and workshops. Our goal is for students completing COMP90051 to gain experience "upskilling" in SML so that they will continue to learn about SML long after their university studies complete. The Project accomplishes this by challenging students with a difficult learning task: predicting authors of test tweets from among a *very large* number of authors found in training tweets. The Project also builds generic skills in problem solving, critical analysis, presentation/communication, and team work – all critical for practical SML.

## Deliverables

1. The predicted labels of the test tweets submitted to the Kaggle in-class competition described below.
2. A report of 3 pages maximum[2] submitted to LMS, assessed and formatted as detailed below.
3. [Not marked[1]] A completed Group Agreement submitted to LMS, please by 29 August 5PM
4. [Not marked[1]] A submission to LMS of your Kaggle team name by 29 August 5PM
5. [Not marked[1]] A zipped archive of code (but please, no data). It need not be all code you wrote, but it should be able to run on lab machines to produce your final Kaggle predictions. You may use any freely available software you like. Submit this to LMS together with your final report.
6. [Not marked[1]] A post-project survey on LMS to understand how effectively teams worked together.

## Assessment Criteria

Kaggle performance: (7 marks), of which:

> Classification accuracy: (5 marks)
> Let *A* be the "categorization accuracy" % on Kaggle over the final private leaderboard (all of test data). Then this portion of your mark is taken to be (max(min(*A*,8),2)-2)*5/6.

[1] The group agreement and post survey may be used in *rare* cases of suspected academic dishonesty or group disagreements. For almost all teams, it is expected that these components will not be used at all by staff. We may however randomly choose teams to run code submissions, to double check results have not been "looked up" – particularly for top-scoring teams.
[2] Any pages past 3 pages will go unmarked. No additional page allowance for references or appendices/supplemental material. Don't waste a page on a cover page. Use font size minimum 10pt with standard fonts.

*Example 1: If you scored 2% (or less) you would receive 0 out of 5*
*Example 2: If you scored 5% you would receive 2.5 out of 5*
*Example 3: If you scored 8% (or more) you would receive 5 out of 5*

Team ranking: (2 marks)
If *N* teams of enrolled students compete, there are no ties and you come in at *R* place (e.g. first place is *R*=1 and last is *R*=*N*) then this portion of your mark is taken to be 2*(*N*-*R*)/(*N*-1). Ties are handled "favourably": a set of tied teams all receive the rank of the highest team as if no teams were tied.
*Example 1: Teams A, B, C, D, E came 1st, 4th, 2nd, 2nd, 5th receive marks 2, 0.5, 1.5, 1.5, 0 out of 2.*

Report quality: (18 marks), of which:

Feature engineering: (4 marks)
You have identified appropriate raw features for identifying authors. There is no one set of features we are looking for, but we expect that to receive top marks for this component, you have done sufficient independent research. More broadly you have done appropriate feature engineering.

Learners: (4 marks)
A project on SML can't be without trying a range of models coupled with training algorithms. While we don't require that your final chosen Kaggle submission uses *any* SML – could just be clever features – you can't *know* this is the best possible approach without trying SML. We expect you to think critically about requirements of the challenge and look into appropriate approaches.

Model selection: (4 marks)
A major trap in applied SML is overfitting. We've learned about this in-depth during lectures and workshops. This component assesses your model selection strategy: how do you choose your final predictions? The public leaderboard is *oh-so-tempting* to use for model selection, but like most real SML problems, you want to generalise!

Critical analysis: (3 marks)
So you've reported on your accuracies and that SimpleBayes does poorly while BeastNets do great. But *why* did you observe these results? Why do you think your features are appropriate? Why should the reader be convinced in your approach? Critical analysis should discuss the why. Sometimes you might draw on known pros/cons of SML approaches to explain your results.

**Presentation quality**: (3 marks)
So you have amazing results, a good understanding of what worked/didn't work. But if we can't follow what you've written – be it grammar, report organisational flow, or incomprehensible plots – then your report could be marked down on presentation quality.

## Why Authorship Attribution?

I'm glad you asked! Authorship attribution has several applications. One is for identifying who wrote an ancient text to better understand its historical context; or similarly in literature, for books written under a pen name. Authorship attribution is closely related to plagiarism detection: did a student really submit this essay, or a professional essay writer? A related area is *stylometric attacks* in privacy research: online authors who might want to write text critical of their government or employer may go to pains to hide their IP address and name, but they should understand that their writing style can give their identity away. Demonstrating that authorship attribution is possible out in the open helps raise awareness to privacy risks of how data is collected, shared and released.[3] Finally, authorship attribution is an interesting application of so-called *xtreme classification* when framed as massively multiclass classification. Our hope is that you will look into some (not necessarily all) of

[3] Not unlike recent UOM work on reidentifying travellers in a 2018 release of myki data. http://a.msn.com/01/en-au/AAFPDcC

these areas in developing your Project submissions. While you might like to use some NLP, we don't expect anything outside COMP90049, COMP90051 and independent research is required.

## Dataset

The data needed for the Project is available on Kaggle. Note, you must abide by rule (R1) below on data use.

> `train_tweets.zip` – contains a single text file representing the training set of over 328k tweets authored by roughly 10k Twitter users. The file is tab-delimited with two columns: (1) a random ID assigned to each user in the dataset; (2) transformed text of a tweet.
>
> `test_tweets_unlabeled.txt` – a text file representing the unlabelled test set of over 35k tweets authored by the same users as in the training dataset. These authors have been removed from the file. It is your task to identify them. An original file of all tweets combined was split with each tweet randomly assigned to test with probability 0.1 and train with probability 0.9 (subject to constraints that users appear in the training dataset). Thus, the test data is an approximately-stratified random sample.

The files are encoded as utf-8, with the tweets transformed somewhat:

- ✓ Whitespace (inc. tabs, newlines) have been converted to spaces, with repeat spaces stripped.
- ✓ Mentions like "@bipr" that would reveal useful user information have been converted to "@handle" so as to preserve some mention statistics.

Some generic pointers about Twitter data: tweets have short, restricted lengths; "RT" sometimes means retweet or in other words resharing of someone else's tweet; besides mentions being special, Twitter also permits hashtags such as #smlftw; images or other media that might have been tweeted have been removed. Other than these peculiarities of Twitter, tweets can be regarded as strings of text.

## Twitter Data Disclaimer

Please note that the dataset is a sub-sample of actual data posted to Twitter, with no filtering whatsoever. Unfortunately, some of the information expressed in the tweets is undoubtedly in poor taste. We would ask you to please look beyond this to the task at hand, as much as possible. The opinions expressed within the tweets in no way express the official views of the University of Melbourne or any of its employees; using the data in a teaching capacity does not constitute endorsement of the views expressed within. Please note also that the data contains links to URLs that may be offensive or malicious in nature. Never follow or retrieve any data from unknown URLs as doing so may expose you to cybersecurity risks. The University accepts no responsibility for offence or harm caused by any content contained within the data.

## Using Kaggle

The Kaggle in-class competition URL: `https://www.kaggle.com/t/cb6ceb3bf96a48819d6b4f0994fb58db`

To participate do the following:

- 🏆 Each team member should create a Kaggle account (unless they have one already)
- 🏆 Create a single Kaggle team noting rule (R2) below on multiple teams
- 🏆 Don't forget to submit your team name through LMS – without a mapping between LMS Group and Kaggle team we can't give you Kaggle-based marks*!*
- 🏆 You may make up to 8 submissions per day ONLY VIA TEAM SUBMISSIONS – see rule (R2). A submission is a comma-separated value (CSV) file with header role "Id,Predicted", with first column populated 1, 2, …, 35437 and second column populated with your test predictions of author user ID. An example submission file can be found on the Kaggle site.

- 🏆 Submissions will be evaluated by Kaggle for accuracy, against just 30% of the test data, forming the public leaderboard
- 🏆 Prior to competition close, you may select a final submission out of the ones submitted previously – by default the submission with highest public leaderboard score is selected by Kaggle
- 🏆 After competition close, public 30% test scores will be replaced with the private leaderboard 100% test scores.

## Report Guidance

As instructed, your report can be at most 3 pages total with standard font size.[2] While you should align with the report scoring details above, as a suggestion only you consider covering the following. You may deviate from this mould: V short intro to problem/data; Description of features, ML methods, model selection referencing literature as appropriate; Summarise and explain results with critical analysis. However you might find other topics useful to cover depending on your approach: engineering/pipeline tricks of interest, future work of ideas that didn't get tried due to time limitations.

## Changes/Updates to the Project Specification

If we require any (hopefully small-scale) changes or clarifications to the project specifications, they will be posted on the LMS. Any addenda will supersede information included in this document.

## Rules and Academic Misconduct

In the Project, students must

(R1) Not use or access outside/extra Twitter data for the Project[4];

(R2) Only make submissions to Kaggle through their team accounts. One and only one team account per group. Do NOT make submissions to Kaggle individually or through "fake" teams in order to get around the daily Kaggle submission limit (doing otherwise is against Kaggle terms of use, and against COMP90051 rules).

(R3) Help their group members out and contribute their fair share. Communicate *early* with your team members – starting with the Group Agreement – if troubles arise.

(R4) Start on your projects early so that any unscheduled problems don't prevent timely submission.

(R5) Have fun and learn to (machine) learn!

For most students, collaboration will form a natural part of the undertaking of this project. However, it is still an individual group task, and so reuse of ideas or excessive influence in algorithm choice and development across different teams will be considered cheating. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy (`http://academichonesty.unimelb.edu.au/policy.html`) where inappropriate levels of collusion or plagiarism are deemed to have taken place.

## Late Submission Policy

As this is a whole-of-class activity, we will not accept late submissions. See (R4) to ensure on time submission. Another suggestion is to submit a draft version on LMS early, so that in the case of "accidents" you will have *something* marked. You can always resubmit your report repeatedly until the deadline. Similarly on Kaggle.

[4] Twitter presents a tempting source of test labels. However, we have chosen a very large number of old test tweets. Too many to manually download; and due to the setup of the Twitter API, not simple to automatically retrieve. *However*, some students might be tempted to look-up tweets from Twitter. Our random code check policy should strongly discourage this. Cheating on the competition will at best achieve a couple of marks more, at worst could result in staff finding students to be cheating. Don't do it.