School of Computing and Information Systems
The University of Melbourne
COMP90049, Introduction to Machine Learning, Semester 2 2024

Assignment 2: Predicting Supreme Court Rulings

Released: Friday, September 6th 2024.

Due: **Stage I**: Friday, October 4th 5pm
Stage II: Wednesday, October 9th 5pm

Marks: The Project will be marked out of 30, and will contribute 30% of your total mark.

1 Overview

In this assignment, you will work with a dataset containing legal cases which were heard in the Supreme Court of the United States. If someone disagrees with the ruling of a lower court, they can appeal it by petitioning to the Supreme Court. In the Supreme Court, the case will be heard and discussed by six to nine judges who can either **reverse** the previous decision (**successful appeal**), or **affirm** (keep) the previous decision (**unsuccessful appeal**).

The assignment is divided into two stages: In Stage 1, you will develop a set of models to address key research questions and summarise your findings in a research paper-style report. You will also participate in a Kaggle in-class competition. In Stage 2, you will review two anonymous submissions from your peers, giving you the opportunity to reflect on different approaches and provide feedback. Throughout the project, you are expected to demonstrate your understanding of machine learning techniques and clearly communicate your knowledge in a report. You are expected to read this specification in full before commencing the project.

1.1 Data and features

You will develop and critically analyse models for predicting the **Supreme Court decision (reverse or affirm)** based on the texts of the oral discussions, as well as metadata about the case and the involved judges. That is, given a case, your model(s) will predict a binary label. You will be provided with a data set of over 5,000 past Supreme Court rulings and their decisions. Each case is represented by different classes of features (Table 1.1).

> **Features x1–x13** contain information that is available **before any decision is made**, i.e., that could be used in a realistic usage situation of the resulting model. When developing your classifiers initially, you should only use (a subset of) these features. (Section 1.2, **RQ1**)
>
> **Features x14–x15** provide information that becomes available only **after or at decision time**. Using this information would be considered cheating. You may use these features to analyse the performance of your classifiers, or biases in the data set. (Section 1.2, **RQ2**)

Features x16–x17 provide personal information about the judges deciding the case. While this information is publicly available, it should **not** influence the decision of the court (or improve you classifier). You may – but do not have to – use these features to explore your classifier performance in-depth. (Section 1.2, **RQ3**)

1.2 Description of tasks

Your overall task is to predict the Supreme Court decision (reverse or affirm) based on the texts of the oral discussions, as well as some metadata about the case. You are strongly encouraged to make use of machine learning software and/or existing libraries in your attempts at this project (such as `sklearn` or `scipy`). You are expected to address the following research questions (RQ):

RQ1 Compare at least ONE baseline (majority, random) and TWO machine learning models in terms of their performance, using (any subset of) features x1–x13. In addition to reporting and comparing the results, you are expected to perform and document steps that ensure the quality of your experiments, such as analyse the data distribution, perform hyper-parameter search, if applicable, and examine if model training leads to overfitting. You are also required to explain the strengths and weaknesses, advantages and disadvantages of the different approaches you tried.

RQ2 Explore whether features x14 and/or x15 affect the performance of the models. For example, you can explore if using features that reflect the difficulty of the case (length of its hearing, vote unanimity etc) can improve the prediction.

RQ3 You must address **one** additional research question yourself, choosing appropriate models, data splits and evaluation methods. We provide two possible RQs for your inspiration, but you are free to choose a different one.

RQ3a: How does Issue Area impact the Supreme Court case rulings? Most cases in the given data set are assigned to one out of 14 issue areas (Feature x8), with a few cases having an UNKNOWN area. Explore the extent to which your model learns features that *generalizes across* issue areas. You will want to compare models that *share* features across areas against models that do not do this. You may want to experiment with predicting missing values for cases with an UNKNOWN issue area, and assess the impact on performance.

RQ3b: Exploring Bias in Supreme Court Predictions Personal attributes of Supreme court judges should not be predictive of the final decision. Explore how features x16-x17 impact your model performance. For example, you can explore if the number of voting judges, or the political orientation of the judges impacts the decision. Or, explore whether courts under different 'chief justices' (x16) exhibited different ruling patterns.

The goal of this assignment is to **critically assess** the effectiveness of various Machine Learning algorithms on the problem of determining the Supreme Court decision, 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. There will be a Kaggle in-class competition where you can compare the performance of your algorithms against your classmates. Note that we expect it to be difficult to achieve substantial performance improvements on this task; among the standard models we tried the best one achieved a gain of only 0.03 in accuracy against the majority baseline. Thus, the goal of the project is to thoughtfully compare and explain models and features rather than achieve major performance gains.

ID	Name	Description
x1	'title'	the name of the case
x2	'petitioner'	the party who appealed the case decision to the Supreme Court
x3	'respondent'	the respondent to that case
x4	'petitioner_state'	the state where the petitioner is located (not all cases have this value)
x5	'respondent_state'	the state where the respondent is located (not all cases have this value)
x6	'petitioner_category'	the category to which the petitioner belongs (state, business, organization, ...)
x7	'respondent_category'	the category to which the respondent belongs (state, business, organization, ...)
x8	'issue_area'	encodes the main area of the law applicable to the case, such as Criminal Procedure, Civil Rights, Privacy, etc
x9	'year'	the year when the case was filed
x10	'argument_date'	the day, month, and year that the case was orally argued before the Court
x11	'court_hearing_-length'	length of the Court discussion regarding the case (in minutes)
x12	'utterances_number'	number of utterances, or turns (i.e. when speakers switch during the conversation) in the Supreme Court discussion
x13	'court_hearing'	the text of the Court's discussion, with utterances separated by five vertical bars (|||||)
x14	'decision_date'	the day, month, and year of the decision
x15	'majority_ratio'	the ratio of judges who voted with the majority vs judges who voted with the minority. Please note that this ratio is about the vote distribution (how unanimous the voting was), and does not reflect the winning side, i.e. the majority of judges can vote to reverse the decision or to keep it.
x16	'chief_justice'	the chief judge of the court when the current case was decided. One of [Burger, Rehnquist, Roberts, Warren]
x17	'justices'	list of the judges who took part in the voting. Each of the judges is represented as a dictionary with the following fields: 'name': the judge's ID, 'born_year': the year of their birth, 'gender': their gender (male, female), 'political_direction': the political direction that best describes the judge's voting behavior (Liberal or Conservative).
Y	'successful_appeal'	0 or 1, shows if the original decision was reversed (1, which means the appeal to the Supreme Court was successful) or affirmed (0, which means that the appeal was unsuccessful and the previous ruling was kept). This is the label we are trying to predict.

Table 1: Input features (x1–x13), information about the decision (x14–x15), sensitive features (x16–x17), and label (Y) of the Supreme Court ruling dataset.

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: Model development and testing and report writing (due October 4th):

1. One or more programs, written in Python, including all the code necessary to reproduce the results in your report (including model implementation, label prediction, and evaluation). You should also include a README file that briefly details your implementation. *Submitted through the LMS.*
2. An anonymous written report, of 2000 words ($\pm 10\%$) **excluding** reference list. Your name and student ID should **not** appear anywhere in the report, including the metadata (filename, etc.). *Submitted through the LMS/Turnitin.* **You must upload the report as a separate PDF file**. Do **NOT** upload it as part of a compressed archive (zip, tar, ...) file or in a different format.
3. Predictions for the test set of court rulings submitted to the Kaggle[1] in-class competition described in Sec 7.

Stage II: Peer reviews (due October 9th):

1. Reviews of two reports written by your classmates, of 150-300 words each. *Submitted through LMS.*

3 Data Sets

You will be provided with

- A *training* set of 4,612 supreme court rulings, with features x1–x17 (Table 1.1) and labelled with the court decision (reversed or affirmed)
- A *development* (validation) set of 577 labeled court rulings, with features x1–x17 (Table 1.1) and labels which you may use for model selection and tuning;
- A *test* set of 577 court rulings, with features x1–x17 (Table 1.1) but with no target labels. This data set will be used for final evaluation in the Kaggle in-class competition.

3.1 Conversation Embeddings

To aid in your initial experiments, we have provided two different representations of the full court conversations (feature x13). You may use any of these representations in your experiments, and you may also engineer your own features from the raw conversations if you wish. The provided representations are:

I. Raw The raw court discussion in plain text is provided in the field `court_hearing` in the raw data *.jsonl files. You may use this field to engineer your own representation of the discussion, for example, use TFIDF vectorisation over the whole text or some particular segments of it.

II. Embedding We mapped each court discussion to a 384-dimensional embedding computed with a pre-trained language model, called the Sentence Transformer (Reimers and Gurevych, 2019).[2] These vectors capture the "meaning" of each court discussion so that similar discussions will be located closely together in the 384-dimensional space. E.g.,

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

[2] https://pypi.org/project/sentence-transformers/

[2.05549970e-02, 8.67250003e-02, 8.83460036e-02, -1.26217505e-01, 1.31394998e-02, . . .]

a 384-dimensional list of numbers

Data format The main data files containing features x1–x17 are provided in JSON lines format (train.jsonl, dev.jsonl and test.jsonl). The labeled data sets also contain the target label (Y).

The **Sentence embedding** representations are provided as dense NumPy matrix (files ending in *.npy).[3]. Line numbers for the same data set type refer to the same instance, e.g., line 5 in train.jsonl and sembed\train.npy are different representations of the same court conversation.

4 Project Stage I

This is the main part of the project, where you are expected to address research questions as explained below, and summarise your finding in a research paper-style report.

4.1 Research Question

You should address **three** research questions in your project, as described in Section 1.2. RQ1 and RQ2 must be approached. For RQ3, you may either choose one of the questions we proposed for inspiration, or propose your own. Your report should clearly state which RQ3 you are addressing. Addressing more than one RQ3 does **not** lead to higher marks. We are more interested in your *critical analysis* of methods and results, than the coverage of more content or materials. However, for RQ1 you should *minimally* implement and analyse in your report **one baseline**, and **at least two different** machine learning models. **N.B.** We are more interested in your **critical analysis** of methods and results, than the raw performance of your models. You may not be able to arrive at a definitive answer to your research questions, which is perfectly fine. However, you should analyse and discuss your (possibly negative) results in depth.

4.2 Feature Engineering (optional)

We have discussed three types of attributes in this subject: categorical, ordinal, and numerical. All three types can be constructed for the given data. Some machine learning architectures prefer numerical attributes (e.g. k-NN); some work better with categorical attributes (e.g. multivariate Naive Bayes) – you will probably observe this through your experiments.

It is **optional** for you to engineer some attributes based on the `raw` court discussions (and possibly use them instead of – or along with – the feature representation provided by us). Or, you may simply use the text features (sentence embeddings) that we generated for you. In addition to (or instead of) text features, you may select to use any combination of categorical, ordinal, and numerical features from attributes x1–x12.

[3] Learn here how to read and process these files: https://numpy.org/doc/stable/reference/generated/numpy.load.html.

4.3 Evaluation

The objective of your learners will be to predict the labels of unseen data. We will use a **holdout strategy**. The data collection has been split into three parts: a training set, a development (validation) set, and a test set. This data is available on the LMS.

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

4.4 Report

You will submit an **anonymised** report of 2000 words in length ($\pm 10\%$), **excluding** reference list. The report should follow the structure of a short research paper, as discussed in the guest lecture on Academic Writing. It should describe your approach and observations in the context of your chosen research question, both in engineering (optional) 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 machine learning models should only be discussed if it is important for connecting the theory to your practical observations.

- Introduction: a short description of the problem and data set, and the research question addressed
- Literature review: a short summary of some related literature, including the data set reference and at least two additional relevant research papers of your choice. You might find inspiration in the Reference list of this document. You are encouraged to search for other references, for example among the articles cited within the papers referenced in this document.
- Method: Identify the newly engineered feature(s), and the rationale behind including them (Optional). Explain the ML models and evaluation metric(s) you have used (and why you have used them)
- Results: Present the results, in terms of evaluation metric(s) and, ideally, illustrative examples. Use of tables and diagrams is highly recommended.
- Discussion / Critical Analysis: Contextualise** the system's behavior, based on the understanding from the subject materials as well as in the context of the research question.
- Conclusion: Clearly demonstrate your identified knowledge about the problem
- A bibliography, which includes Fang et al. (2023b), 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.

** Contextualise implies that we are more interested in seeing evidence of you having thought about the task, and determined *reasons* for the relative performance of different methods, rather than the raw scores of the different methods you select. 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.

We will provide LaTeXand 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.

5 Project Stage II

During the reviewing process, you will read two anonymous submissions by your classmates. This is to help you contemplate some other ways of approaching the Project, and to ensure that every student receives some extra feedback. You should aim to write 150-300 words total per review, responding to three *'questions'*:

- Briefly summarise what the author has done in one paragraph (50-100 words)
- Indicate what you think that the author has done well, and why in one paragraph (50-100 words)
- Indicate what you think could have been improved, and why in one paragraph (50-100 words)

6 Assessment Criteria

The Project will be marked out of 30, and is worth 30% of your overall mark for the subject. The mark breakdown will be:

Report Quality: (26/30 marks)
You can consult the marking rubric on the LMS/Assignment 2 page which indicates in detailed categories what we will be looking for in the report.

Kaggle: (2/30 marks)
For submitting (at least) one set of model predictions to the Kaggle competition. Your marks will not depend on your results in this competition.

Reviews: (2/30 marks)
You will write a review for each of the two reports written by other students; you will follow the guidelines stated above.

7 Using Kaggle

Task The Kaggle competition will be on predicting the results of the Supreme Court hearing: **reversed** or **affirmed**.

Instructions The Kaggle in-class competition URL will be announced on the LMS shortly. To participate do the following:

- 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 30% of the test data, forming the public leaderboard.
- Prior to the closing of the competition, 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 closes, public 30% test scores will be replaced with the private leaderboard 100% test scores.

8 Assignment Policies

8.1 Terms of Data Use

The data set is derived from the resource published in Fang et al. (2023b):

> Biaoyan Fang, Trevor Cohn, Timothy Baldwin, and Lea Frermann. 2023. Super-SCOTUS: A multi-sourced dataset for the Supreme Court of the US. In Proceedings of the Natural Legal Language Processing Workshop 2023, pages 202–214, Singapore. Association for Computational Linguistics.

This reference **must** be cited in the bibliography. We reserve the right to mark any submission lacking this reference with a 0, due to violation of the Terms of Use. We include other related references in the References section, in the end of this document.

Changes/Updates to the Project Specifications

We will use LMS announcements for any large-scale changes (hopefully none!) and Ed for small clarifications. Any addendums made to the Project specifications via LMS will supersede information contained in this version of the specifications.

Late Submission Policy

We allow **no extensions or late submissions** to ensure a smooth peer review process. Submission will close at **5pm on October 4th**. Students who are eligible for **special consideration** (e.g., with an APP) please email Lea Frermann (lea.frermann@unimelb.edu.au>) and a solution will be accommodated.

Academic Misconduct

For most students, discussing ideas with peers will form a natural part of the undertaking of this project. However, it is still an individual task, and so reuse of ideas or excessive influence in algorithm choice and development will be considered cheating. We highly recommend to (re)take the academic honesty training module in this subject's LMS. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy[4] where inappropriate levels of collusion or plagiarism are deemed to have taken place. Content produced by generative AI (including, but not limited to, ChatGPT) is *not* your own work, and submitting such content will be treated as a case of academic misconduct, in line with the University's policy.

[4] http://academichonesty.unimelb.edu.au/policy.html

References

Fang, B., Cohn, T., Baldwin, T., and Frermann, L. (2023a). More than votes? voting and language based partisanship in the US Supreme Court. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4604–4614, Singapore. Association for Computational Linguistics.

Fang, B., Cohn, T., Baldwin, T., and Frermann, L. (2023b). Super-SCOTUS: A multi-sourced dataset for the Supreme Court of the US. In Preoțiuc-Pietro, D., Goanta, C., Chalkidis, I., Barrett, L., Spanakis, G., and Aletras, N., editors, *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 202–214, Singapore. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.