Contents



1. IN5550 Obligatory 1 common mistakes

Results reporting



Common mistakes

- Drew conclusions from experiments that don't have any results reported
  - eg. saying something like *'I experimented with multiple layers and found that 3 is the best choice'* — but there is nothing reported with different number of layers
- Did not organize the results in tables and used screenshots of terminal output or numbers scattered through text instead
- Did not number tables
- Used abbreviations or made up names for experiment set ups or models in tables but never introduced what they mean

Results reporting

Do's and dont's

Do's

- Put your results in numbered tables with clear column and row names.
- Discuss your results by clearly referring to the relevant table.
- Clearly introduce any used abbreviations/made up names.
- Store large files in the project work directory on Fox (`/cluster/work/projects/ec403/`).

Dont's

- Report results directly in text or in screenshots
- Use unnumbered tables, cryptic columns and rows
- Store files (models, datasets, etc) in your home directory on Fox (which we cannot access, of course)

Evaluation script



Common mistakes

- No evaluation script provided
- Evaluation script that doesn't work (runtime errors)
- Evaluation script that attempts to read the training set and start training
- Evaluation script creating the label encoder entirely from scratch at the test time from the test set. It means that the textual class labels can be mapped to different integer values than during the training.

Evaluation script



Do's and dont's

Do's

- Provide an evaluation script as required and with running examples
- Make sure it can be run on a different set than the one it was trained on
- Load everything that's needed (model, training vocabulary, etc.)
- Include the output of sklearn `Classification Report()`

Dont's

- Train the model again at test time
- Map words/labels to different integers at test time
- Fit your tokenizer again from scratch at test time

Words preprocessing

Common mistakes

- Always lower-casing the input text before feeding it to the word embedding model.
  - Not all word embeddings use lower-cased tokens!! For example, models 10 and 222 from `https://vectors.nlpl.eu/repository/` and on Fox preserve the words case

Do's and dont's

Do's

- Make sure the words are in the format expected by the word embedding model you are loading.
- Always look at the data, be creative about pre-processing and cleaning (e.g., remove digits)

Dont's

- Use the same preprocessing method regardless of what the word embedding model takes as input.

Code cleanliness

Common mistakes

- No repository organization:
- Classes and functions duplicated in several different scripts across the repository

Do's and dont's

Do's

- Use `.py` scripts instead of Jupyter notebooks.
- Write clean code (well-organized, good names, documentation etc.).

Dont's

- Duplicate code.
- Automatically generate `.py` scripts from notebooks without cleaning it up.
- Forget to provide a link for your Github repository.

Training

Common mistakes

- Use early-stopping!

Do's and dont's

Do's

- Use a training stopping criterion (early-stopping)
- use stratified sampling and weighting the target classes (can be helpful)

Dont's

- tune the number of epochs as a hyper-parameter
- decompress the datasets manually; `pandas` can read and write compressed files

More mistakes

Some more mistakes

- Not giving access to all the teachers (you can find us at `https://github.uio.no/orgs/in5550/teams/2025`)
- Putting excuses/disclaimers in your report.
- Not addressing explicit tasks from the obligatory text.
- Inconsistency in results reporting between different parts of the report (eg. using different metrics, f1-score in one part and accuracy in another)

More mistakes

Some more mistakes

- Not giving access to all the teachers (you can find us at `https://github.uio.no/orgs/in5550/teams/2025`)
- Putting excuses/disclaimers in your report.
- Not addressing explicit tasks from the obligatory text.
- Inconsistency in results reporting between different parts of the report (eg. using different metrics, f1-score in one part and accuracy in another)
- Please always declare the full composition of your team when submitting the assignment in Devilry (all the team members!)
- NB: 'Gensim' is not a name of a word embedding algorithm, it's a piece of software!