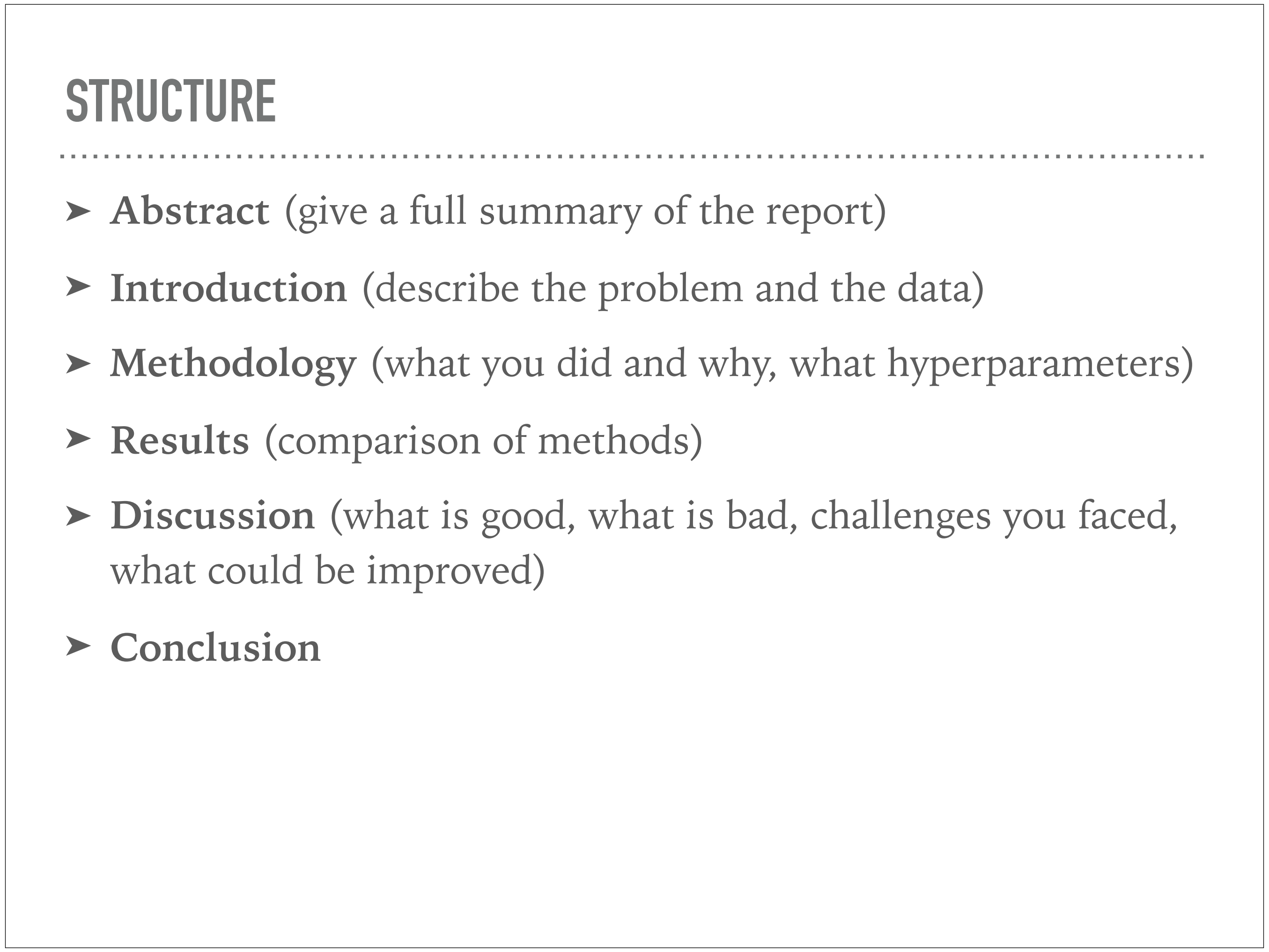
ML PAPER WRITING: BEST PRACTICE

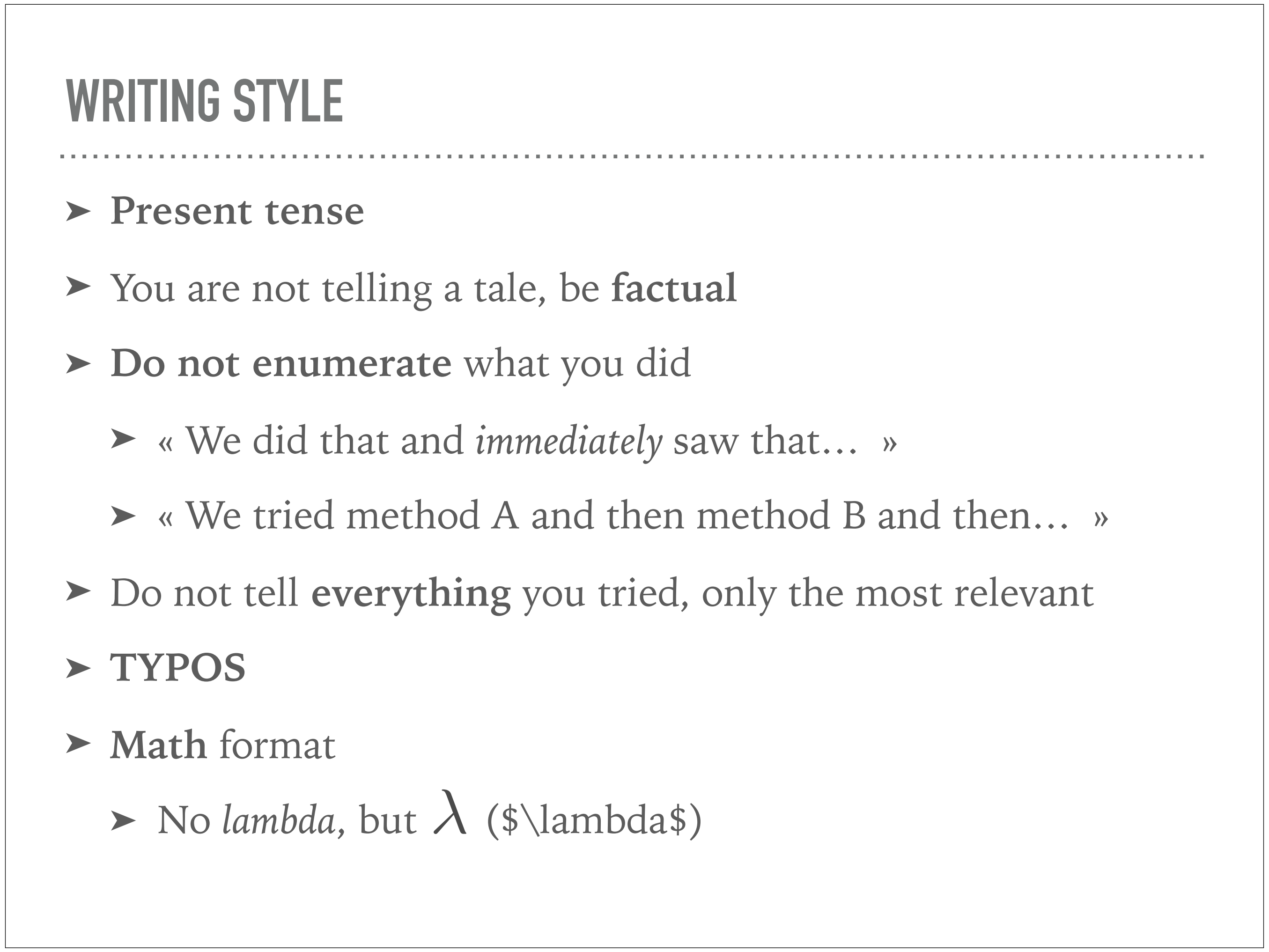
PCML16 - Project 1



STRUCTURE

- **Abstract** (give a full summary of the report)
- **Introduction** (describe the problem and the data)
- **Methodology** (what you did and why, what hyperparameters)
- **Results** (comparison of methods)
- **Discussion** (what is good, what is bad, challenges you faced, what could be improved)
- **Conclusion**

Most of you did that correctly
Non exhaustive



WRITING STYLE

- **Present tense**
- You are not telling a tale, be **factual**
- **Do not enumerate** what you did
  - « We did that and *immediately* saw that… »
  - « We tried method A and then method B and then… »
- Do not tell **everything** you tried, only the most relevant
- **TYPOS**
- **Math** format
  - No *lambda*, but λ (`$\lambda$`)

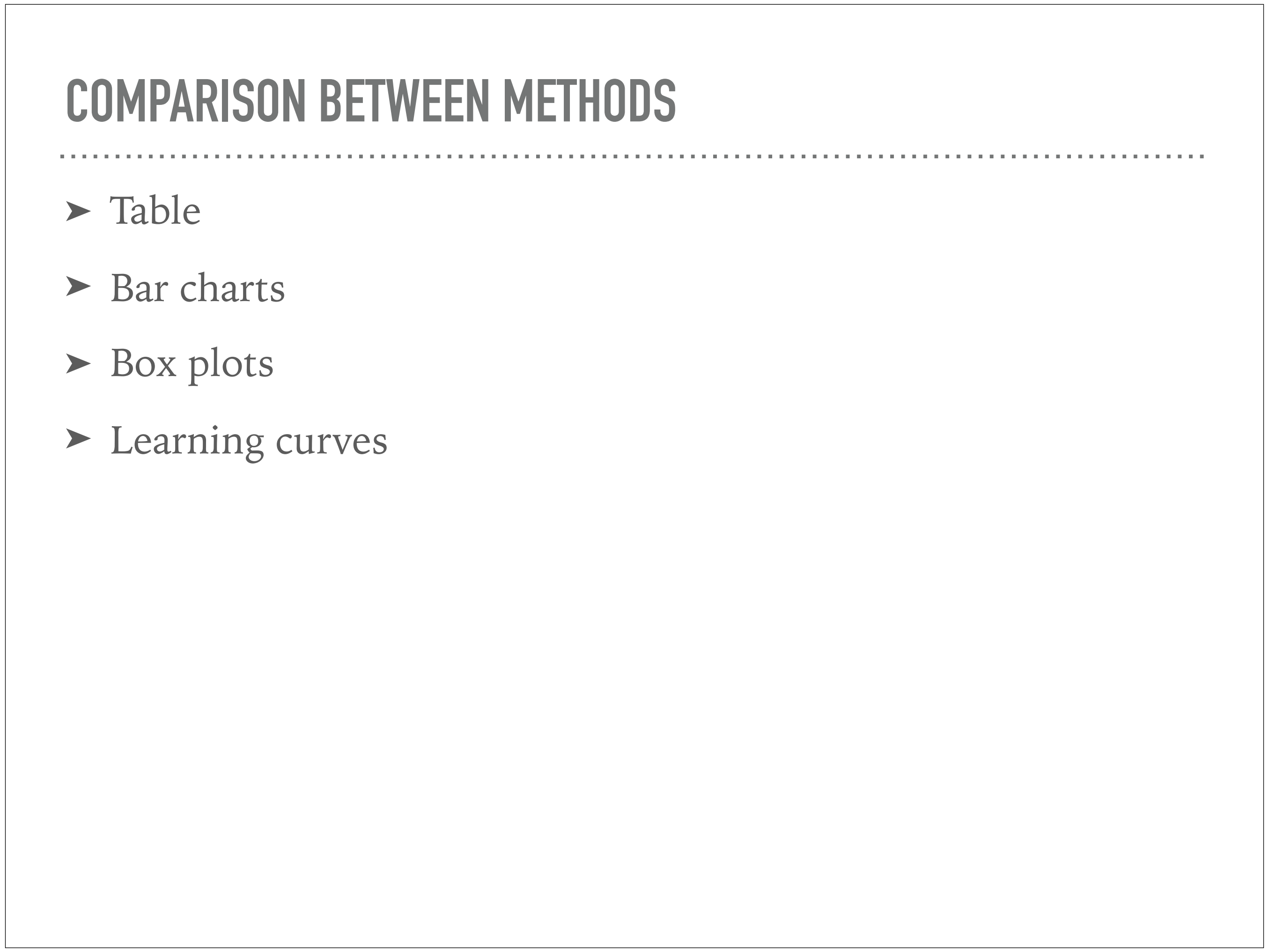
It is not because we are doing science that we should neglect language. Communication is key to convey your ideas and convince your boss, the conference.
Not telling a tale: « The discovery of the Higgs boson has been a great step in the history of the physics discoveries. Although its existence has been acknowledged, its detection is not so easy. In fact it is not observed directly, through lateral factors though. That is why the CERN proposed a machine learning challenge to invite **all the passionates, and those who just love machine learning,** to try to predict the detection of t**his little particle**. In this project we are going to try to make possible the observation of our **beloved** Higgs boson. »

REPRODUCIBILITY

- The reader should be able to **reproduce** what you did
- Give the v**alue** of the hyperparameters
- **Pre-processing** (what you did on train and test set)
- **Feature engineering**
- Don’t **derive** the basic methods again
- Rather describe **your contribution** or non-trivialities

RESULTS

- Start with a **baseline**
  - Random
  - Majority class
  - Constant predictor
  - Any other domain-specific baseline
- Try to **improve** it with a slightly more complex method
- **Repeat**
- **Report** your findings with a table or a plot



COMPARISON BETWEEN METHODS

- Table
- Bar charts
- Box plots
- Learning curves

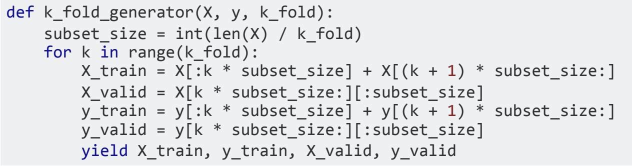
Show good examples

EQUATIONS

- Only if **necessary**:
  - Non-trivialities (metrics, optimization tricks, …)
  - **Your contribution**
- Numbered only if **referred** to if later
- It is **part of the text**, so add **punctuation**
- Example:

CROSS-VALIDATION

- Estimation of **test error** (validation error)
- Useful for:
  - **Comparison** between methods
  - Tuning of **hyperparameters**
- Mean and **standard deviation**
- Python code snippet to generate *k*-fold C-V:



```
def k_fold_generator(X, y, k_fold):
    subset_size = int(len(X) / k_fold)
    for k in range(k_fold):
        X_train = X[:k * subset_size] + X[(k + 1) * subset_size:]
        X_valid = X[k * subset_size:][:subset_size]
        y_train = y[:k * subset_size] + y[(k + 1) * subset_size:]
        y_valid = y[k * subset_size:][:subset_size]
        yield X_train, y_train, X_valid, y_valid
```

PLOTS: A PICTURE IS WORTH A THOUSAND WORDS

- Labels
- Legend
- (Title)
- Ticks
- Caption
- Use different colors
- Use different markers (printed black and white, colorblind people)
- Line size
- Font size
- Scale of your axis (negative accuracy, does it show something interesting)

RECOMMENDED READINGS

- By *recommended*, I mean **mandatory**

CODE

- **PEP8**
- **Modularize** your code
- **TRY IT BEFORE SUBMITTING**
- If something takes long, give **feedback:**
  - **Print** some text
  - Display some **progress bar**
- **README.md** (use MarkDown):
  - Description of the **projects**
  - Description of the **structure** and the **files**
  - Instructions to **run** your code (your grand-mother should be able to do it)