Supervised Learning in Practice

March 5, 2026

Announcements

- **Assignments**
  - **Final Project Checkpoint 1** (1-page proposal) due **TOMORROW at 11:59pm**
  - **Lab 3** due Monday, March 16 at 11:59pm
  - **Lab 4** due Friday, March 27 at 11:59pm
  - **Final Project Checkpoint 2** (code review) due Friday, April 3
- **No in-person class on Thursday, March 19**
  - In lieu of in-person lecture, I will be posting video lecture notes on how to create your own R/Python software package
  - No office hours Wednesday, March 18 and Friday, March 20

Announcements

Register for DataFest 2026: https://vweber1.wixsite.com/datafest-nd

Week after spring break: parallelization

- **Please register for an ND Center for Research Computing (CRC) account: https://docs.crc.nd.edu/new_user/obtain_account.html**
  - Use Safari/Edge if Chrome doesn't work
  - Mention this class if you are asked to justify why you need access to CRC
- **Download either VS Code or Positron** in order to follow along during "Remote SSH" tutorial

Poll: Challenges in Building ML Models

Which parts of the ML model building process have been the most challenging and cumbersome in practice?

Poll: Challenges in Building ML Models

Which parts of the ML model building process have been the most challenging and cumbersome in practice?

My opinion:

- The Data: “Garbage In, Garbage Out”
- What Model?
- Which Hyperparameters?
- Feature Engineering
- Data Splitting and Evaluation

Poll: Challenges in Building ML Models

Which parts of the ML model building process have been the most challenging and cumbersome in practice?

My opinion:

- The Data: “Garbage In, Garbage Out”
- **What Model?**
- **Which Hyperparameters?**
- **Feature Engineering**
- Data Splitting and Evaluation

Today’s plan: Recent Developments in Supervised ML

1. **AutoGluon: AutoML**
2. **TabPFN**

AutoML

AutoGluon: AutoML Toolkit

- **Motivation:** often difficult to find the single "perfect" model through endless hyperparameter tuning
- **AutoGluon:** attempts to automate the process of model building and hyperparameter tuning (in very few lines of code)

```
0  from autogluon.tabular import TabularPredictor

1  model = TabularPredictor(label="Y")
2  model.fit(train_data, presets="medium_quality")
3  y_predictions = model.predict(valid_data)
```

- More on AutoGluon: https://auto.gluon.ai/stable/index.html

AutoGluon: AutoML Toolkit

- **Behind the scenes of `model.fit(...)`**
  1. Training a variety of **different models** (including tree ensembles and neural networks)
  2. Using **bagging** when training those models
  3. **Stack-ensembling** those models to combine their predictive power
- Different presets: “medium”, “good”, “best”, etc (see here)
- How it works: https://auto.gluon.ai/stable/tutorials/tabular/how-it-works.html

TabPFN

TabPFN: Tabular Prior-data Fitted Network [Hollmann et al. (2025)]

- **Foundation Model:** “general purpose” model (in contrast to “task-specific” models)
  - A single, massive model trained on a vast amount of data (text, images, or code) that can be adapted (fine-tuned) to a wide range of downstream tasks.
- **TabPFN:** state-of-the-art **foundation model** for tabular datasets

```
from tabpfn import TabPFNRegressor (or TabPFNClassifier)

model = TabPFNRegressor()
model.fit(X_train, y_train)
y_predictions = model.predict(X_valid)
```

- More on TabPFN: https://www.nature.com/articles/s41586-024-08328-6

TabPFN: Tabular Prior-data Fitted Network [Hollmann et al. (2025)]

- TabPFN was **pre-trained on ~130 million synthetic datasets** generated by causal models
- Underlying architecture is a **transformer** (i.e., the “T” in ChatGPT)
- **In-Context Learning:** feed the model your *entire* training set (X, y) + the test X in a single “prompt"
- **No hyperparameter optimization**
- **Natively handles common data challenges** such as missing values, outliers, and categorical features without manual data preprocessing

TabPFN: Tabular Prior-data Fitted Network [Hollmann et al. (2025)]

- First Timers: https://docs.priorlabs.ai/how-to-access-gated-models
  - Register for huggingface account
  - Accept terms and conditions
  - Create access token
- Note: if you use tabPFN, conda lock most likely won't work across platforms
  - Use one of:
    - conda lock -p osx-arm64
    - conda lock -p osx-64
    - conda lock -p linux-64
    - conda lock -p win-64
- **DO NOT PUSH FILE WITH ACCESS TOKEN IN IT!**

Don’t forget

- **Final Project Checkpoint 1** (1-page proposal) due **TOMORROW at 11:59pm**
- **Please register for an ND Center for Research Computing (CRC) account: https://docs.crc.nd.edu/new_user/obtain_account.html**
  - Use Safari or if Chrome doesn’t work
  - Mention this class if you are asked to justify why you need access to CRC
- Download either VS Code or Positron in order to follow along during “Remote SSH” tutorial