References

- (GBC) Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016) Deep Learning, MIT Press.
- (HTF) Hastie, T., Tibshirani, R. and Friedman, J. (2009) The Elements of Statistical Learning, 2nd edition, Springer.
- (Molnar): Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Christoph Molnar.
- (RW) C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006.

Optional readings

The readings below are purely optional, but are helpful. If you need further explanation of anything covered in lecture, the best way to find an answer is to ask at Piazza (preferred)/email me/see me during my consultation hour.

7. Ensemble methods
    - Review paper on ensemble methods: Lior Rokach (2010). "Ensemble-based classifiers". In: *Artificial Intelligence Review* 33.1-2, pp. 1–39
    - Bagging: HTF 8.7.
    - Random forest: HTF 15.
    - Boosting: HTF 11.
8. Multi-layer perceptrons
    - Gradient-based learning: GBC 4.3, 5.9, 5.10
    - MLP: GBC 6
    - Autodiff: Wikipedia article
    - Introduction to PyTorch: https://pytorch.org/tutorials/beginner/basics/intro.html
9. Convolutional neural networks
    - GBC 9.1-9.5, 9.10
    - LeNet5: Yann LeCun et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324
10. Robust machine learning
    - RANSAC
        - Wikipedia article: https://en.wikipedia.org/wiki/Random_sample_consensus.
        - See sklearn source code for an example implementation: https://github.com/scikit-learn/scikit-learn/blob/844b4be24/sklearn/linear_model/_ransac.py#L56. Note that sklearn implementation and the pseudocode in the Wikipedia article are not exactly the same.
    - Theil-Sen:
        - Wikpedia article: https://en.wikipedia.org/wiki/Theil%E2%80%93Sen_estimator. This describes the Theil-Sen estimator for simple linear regression.
        - Theil-Sen estimators for multiple linear regression: Xin Dang et al. (2008). "Theil-sen estimators in a multiple linear regression model". In: *Olemiss Edu*.

  - See sklearn source code for an example implementation: https://github.com/scikit-learn/scikit-learn/blob/844b4be24/sklearn/linear_model/_theil_sen.py#L207.
- M-estimators: https://en.wikipedia.org/wiki/Robust_statistics#M-estimators
- LAD: https://en.wikipedia.org/wiki/Least_absolute_deviations
- Huber loss: https://en.wikipedia.org/wiki/Huber_loss.
- Huber regressor in sklearn: https://scikit-learn.org/stable/modules/linear_model.html#huber-regression. Note that the Huber regressor discussed in lecture is a simpler version where $\sigma = 1$, and $\alpha = 0$.
- Adversarial examples: see a high-level discussion with links to some papers in this OpenAI blog https://openai.com/blog/adversarial-example-research/.

11. Interpretable machine learning

    - Molnar: 3 (overview of interpretability), 5.1 (linear regression), 5.2 (logistic regression), 5.4 (decision tree, Gini importance), 8.5 (permutation importance), 8.6 (surrogate models).
    - Permutation importance: https://scikit-learn.org/stable/modules/permutation_importance.html.
    - sklearn.inspection.permutation_importance: https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html#sklearn.inspection.permutation_importance.
    - Gini importance for sklearn's RandomForestRegressor: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor.feature_importances_, sklearn's RandomForestClassifier https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClas feature_importances_.

12. Bayesian learning

    - RW: 1, 2, 4, 5
    - sklearn Gaussian processes: https://scikit-learn.org/stable/modules/gaussian_process.html.