Assignment L01 – ML Roadmap

Contributors: Martin Demel, Jiri Musil, Jasmine Campos, Munachimso Benson

Department of Science, Technology, Engineering & Math, Houston Community College

ITAI - 1371 Introduction to Machine Learning

Professor Vishwa Rao

August 24th, 2025.

# Introduction

As a team, we watched a YouTube video, 2020 Machine Learning Roadmap (87% valid for 2024). The presenter defines machine learning practically as turning data into numbers and finding patterns in those numbers using math. He compares this with traditional “software 1.0,” where rules are manually coded, and “software 2.0,” where rules are derived from many labeled examples. We appreciated the hands-on explanation, the playful mind map created in Whimsical, and the focus on learning through exploring links, slides, and a companion GitHub repository with a full image of the map and timestamps for easy navigation.

# Body

## The most effective approach was a disciplined entry rule. If clear rules solve the problem, implement them without using machine learning. Use ML when rules would be too numerous, environments keep changing, or insights need to be extracted from large data sets. The cooking metaphor clarified this. In traditional programming, we specify the recipe. In ML, we gather many input–output pairs, convert them to numbers, and let a model infer the recipe.

## Situations that merit ML

* Long, brittle rule lists where enumerating behavior is impractical.
* Continually changing environments that require adaptation.
* Discovering patterns in large data that are not obvious by inspection.

## Common task families and how to judge them

* Classification. Binary, multi-class, and multi-label tasks are evaluated with a confusion matrix that distinguishes true and false positives and negatives.
* Regression. Involves predicting numerical values and is assessed using MAE, MSE, and R², with the reminder that MSE emphasizes larger errors.
* Sequence-to-sequence. Converts one sequence into another, such as in translation or speech-to-text applications.
* Clustering and dimensionality reduction. Used for grouping similar data and reducing complexity to address the curse of dimensionality.

The presentation treats ML as a loop rather than a model in isolation. It starts with the data and returns there after deployment.

Process, not models, is the centerpiece. The loop begins with data collection. Define the problem, inventory what already exists, respect privacy, and distinguish structured tables from unstructured images, audio, video, and text. Data preparation follows with exploratory data analysis to name features and targets, quantify missingness and outliers, and record a data dictionary so column meanings are unambiguous. Preprocessing then makes everything numeric through imputation, categorical encoding, and scaling or standardization when features live on very different ranges. A three-way split keeps the final check untouched, with proportions cited as about 70% – 80% train, 10% – 15% validation, and 10% – 15% test.

Training is presented as a series of controlled experiments. Select an algorithm that fits the task, starting with familiar estimators and progressing to neural networks when representation learning is necessary. Watch for underfitting and overfitting, then address them with regularization techniques like L1 or L2 penalties, dropout, early stopping, data augmentation, and batch normalization. Adjust hyperparameters carefully, especially learning rate, batch size, model depth or width, number of trees, and number of iterations.

Evaluation, serving, and retraining complete the loop. The presenter emphasizes looking beyond headline metrics to practical constraints like training cost and inference latency. A large production example highlights scale, mentioning approximately 70,000 GPU hours required to train a model.

## Tools and working style

The roadmap emphasizes practical tools over a catalog. Notebooks facilitate exploration. TensorFlow and PyTorch cover deep learning, while scikit‑learn supports traditional methods. Experiment tracking is recommended to prevent loss of results and configurations, with dataset and model versioning highlighted as important. Pretrained model hubs are encouraged to speed up transfer learning. For sharing results with stakeholders, a straightforward user interface is suggested so people can interact with predictions.

## Mathematics and learning approach

The essentials are clear. Linear algebra, multivariate calculus with the chain rule, probability and distributions, and optimization are the core concepts. The learning approach involves starting to build, consulting the math when necessary, and revisiting topics with better questions. The presenter emphasizes the idea of beginning as “cooks” who explore and only later becoming “chemists” who formalize.

# Conclusion

As a team, we learned to think systemically. The value lies in the loop, not just a single notebook. We will prefer rule-based solutions when they are enough, then move to ML for long rule lists, changing environments, and large-scale pattern discovery. Specifically, we will maintain a clean split of 70% – 80% for training, 10% – 15% for validation, and 10% – 15% for testing, reserving the test set for the final check, monitoring imbalance, and matching metrics to costs. We will plan regularization and hyperparameter sweeps in advance and allocate resources for compute and latency early. The presentation’s compass mindset, reinforced by the autonomy example with eight cameras, radar, and a cited 70,000 GPU-hour training scale, gave us a practical template to follow.

**References:**

Daniel Bourke. (2020, July 12). 2020 Machine Learning Roadmap (87% valid for 2024) [Video]. YouTube. <https://www.youtube.com/watch?v=pHiMN_gy9mk>