Introduction to Machine Learning

Ichiro Fujinaga
Schulich School of Music
McGill University

These slides were created with the aid of LLMs.

What is Machine Learning

- Machine learning is a subset of artificial intelligence (AI) that enables computers to learn patterns and make decisions without explicit programming.
- Includes Deep Learning methods including Large Language Models (LLMs).
- ML spans hand-engineered features to learned representations.
- It's like teaching computers to learn and improve from experience.
- It’s adaptive.

Types of Machine Learning

- Supervised Learning
  - The algorithm is trained on a labeled dataset, where each input has a corresponding output label (ground truth).
  - Example: Predicting house prices based on features.
- Unsupervised Learning
  - The algorithm tries to find patterns or groupings in unlabeled data.
  - Examples: Grouping similar customer behaviours. Clustering.
- Reinforcement Learning
  - Learning through trial and error by interacting with an environment, receiving feedback in the form of rewards or penalties with the goal of maximizing cumulative reward over time.
  - Examples: AlphaGo, autonomous driving, and DeepSeek

What is Deep Learning

- Subfield of machine learning based on multi-layer neural networks
- Learns hierarchical representations directly from data
- Reduces need for manual feature engineering
- Enabled by:
  - Large datasets
  - GPUs / accelerators
  - Backpropagation + stochastic optimization

What is the relationship between Machine Learning and AI?

- Most AI algorithms involve learning
  - Machine Learning
    - Deep Learning (representation learning)
  - Rule-based methods (humans learn first)
    - Expert systems, traditional NLP (Natural Language Processing)
- Other algorithms in AI
  - Searching
    - Depth-first, Breadth-first, A*, etc.
  - Optimizations
    - Evolutionary Computing (Genetic Algorithms)

Key Terminology

- Features and Labels
  - Features are input variables and labels are the desired output.
- Training Data
  - Contains features and labels.
  - Teaching machines with examples.
- Model
  - A model is a learned algorithm used for predictions.
- Deep Learning terminology
  - Parameters (weights, biases)
  - Architecture (depth, width)
  - Loss function
  - Backpropagation

How Machine Learning is Built

- Data Collection
  - Quality data is crucial for effective machine learning.
- Training
  - Machines learn patterns from labeled data during training to create a model.
  - In Deep Learning: training may involve millions/billions of parameters.
  - Often pretrained, then fine-tuned.
- Testing and Evaluation
  - Test the model's performance with new data.

Data Splits

- Data is usually divided into two or three groups
  - Training, Validation, and Testing
  - A common split is 70-80% for training, 10-15% for validation, and 10-15% for testing.
- K-fold Cross Validation

Data Splits: Training

- Purpose: To train the machine learning model. During the training phase, the model learns patterns and relationships between input features and output labels.
- Usage: The model is exposed to the training data, and its parameters (weights in neural networks, coefficients in the case of linear models, etc.) are adjusted iteratively to minimize the difference between predicted outputs and actual labels.

Data Splits: Validation

- Purpose: The validation or evaluation data is used during the training phase to assess the model's performance and make adjustments to hyperparameters (e.g., learning rate and regularization).
- Usage: After each training iteration (epoch), the model's performance is evaluated on the validation data.
- Hyperparameter Tuning: The validation set is crucial for tuning hyperparameters to optimize the model's performance without overfitting.

Data Splits: Testing

- Purpose: The testing data is reserved for evaluating the model's final performance after training and hyperparameter tuning. It provides an unbiased assessment of how well the model is expected to generalize to new, unseen data.
- Usage: The model is applied to the testing data, and its predictions are compared to the actual labels.
- Unseen Data: The model should *never* be exposed to testing data during training or validation.

Overfitting / Underfitting

- When the model does not capture the underlying patterns.
  - Overfitting (model too complex?)
    - Low training error (fits the training data very well).
    - High testing error (poor performance on new data).
  - Underfitting (model too simple?)
    - High training error (fails to fit the training data well).
    - High testing error (poor performance on new data).

Solutions to Overfitting / Underfitting

- Increase data size
  - Collect more data
  - Create more data (data augmentation)
- Feature engineering (selection and weighting)
- Hyperparameter fine tuning (modify the model)
- Cross validation
- Deep learning
  - Regularization (dropout, weight decay)
  - Early stopping
  - Pretraining + fine-tuning

Cross Validation

- K*l-fold cross validation
  - https://ich.music.mcgill.ca/classes/mumt621_24/classifiers/trainingDatasets3.pdf
- Leave-One-Out Cross Validation (LOOCV)
  - Made for k-NN!

Responsible AI (1)

- Data privacy
  - Implement robust personal data protection measures, adopting privacy-preserving techniques, to avoid potential misuse or mishandling of sensitive information.
- Transparency and explainability
  - Develop methods to make AI systems more interpretable and providing explanations for their decisions to enhance transparency and user trust.
- Bias and fairness / Inclusivity and accessibility
  - Ensure fairness in AI by addressing bias in training data, algorithms, and decision-making processes to avoid discriminatory behaviour by considering diverse user demographics and addressing accessibility requirements.

Responsible AI (2)

- Ethical considerations
  - Establish ethical guidelines and frameworks promoting responsible use in various applications, such as surveillance, social scoring, or influencing public opinion.
- Environment impact
  - Training complex AI models (e.g., large neural networks), can have a significant environmental impact due to high computational requirements.
  - Develop energy-efficient AI algorithms.

Popular Supervised Learning Algorithms

- Classical ML
  - k-Nearest Neighbour
  - Support Vector Machines
  - Random Forest
  - Gradient Boosting
- Deep Learning
  - Feed-forward neural networks
  - Convolutional Neural Nets
  - Recurrent Neural Nets / Long-Short Term Memory (LSTM)
  - Transformers (Large Language Models)

What Makes LLMs Different?

- Based on transformer architectures
- Trained using self-supervised learning
  - Predict next token, masked token, etc.
- Learn general-purpose representations
- Can be adapted via:
  - Prompting
  - Fine-tuning
  - Retrieval-Augmented Generation (RAG)

How LLMs Change ML Workflows

- Classical ML: task-specific models + datasets
- Deep Learning: end-to-end learning from raw signals
- LLMs: model + prompt + external tools
- Emergence of:
  - Prompt engineering
  - External tool use
  - Human-in-the-loop workflows

Tools

- scikit-learn: Python
- TensorFlow (Google): Python, C++
- PyTorch (Meta): Python, C++
- Hugging Face (models, datasets)
- OpenAI / Anthropic APIs (LLMs)
- Vector databases (FAISS, Milvus)

Simple Classical Classifiers

- Decision trees
- Linear discriminant analysis

Three Iris Species

(https://commons.wikimedia.org/w/index.php?curid=248095)
(https://www.datacamp.com/tutorial/machine-learning-in-r)

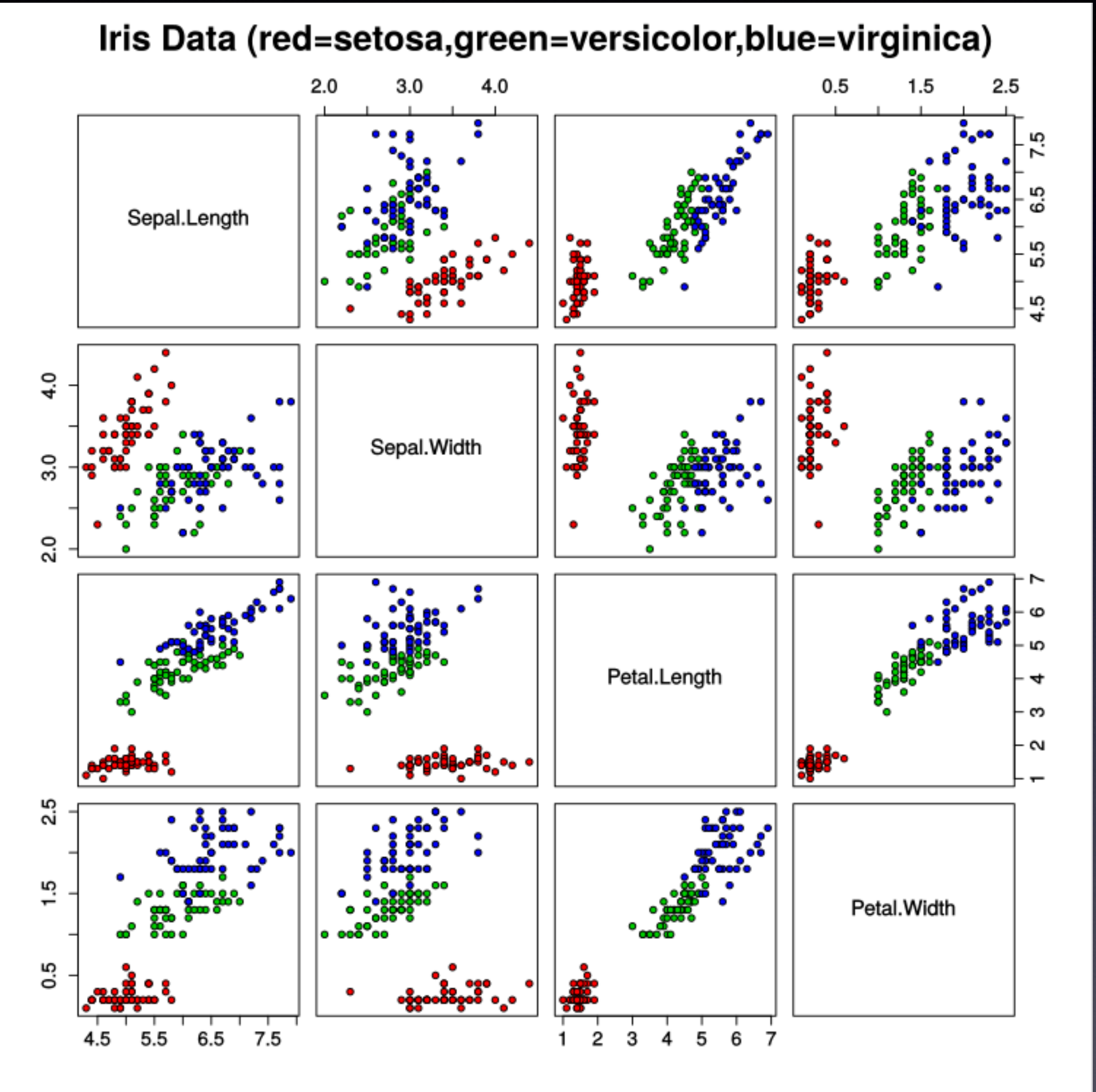
Iris Dataset (1935)

(https://rpubs.com/vidhividhi/irisdataeda)

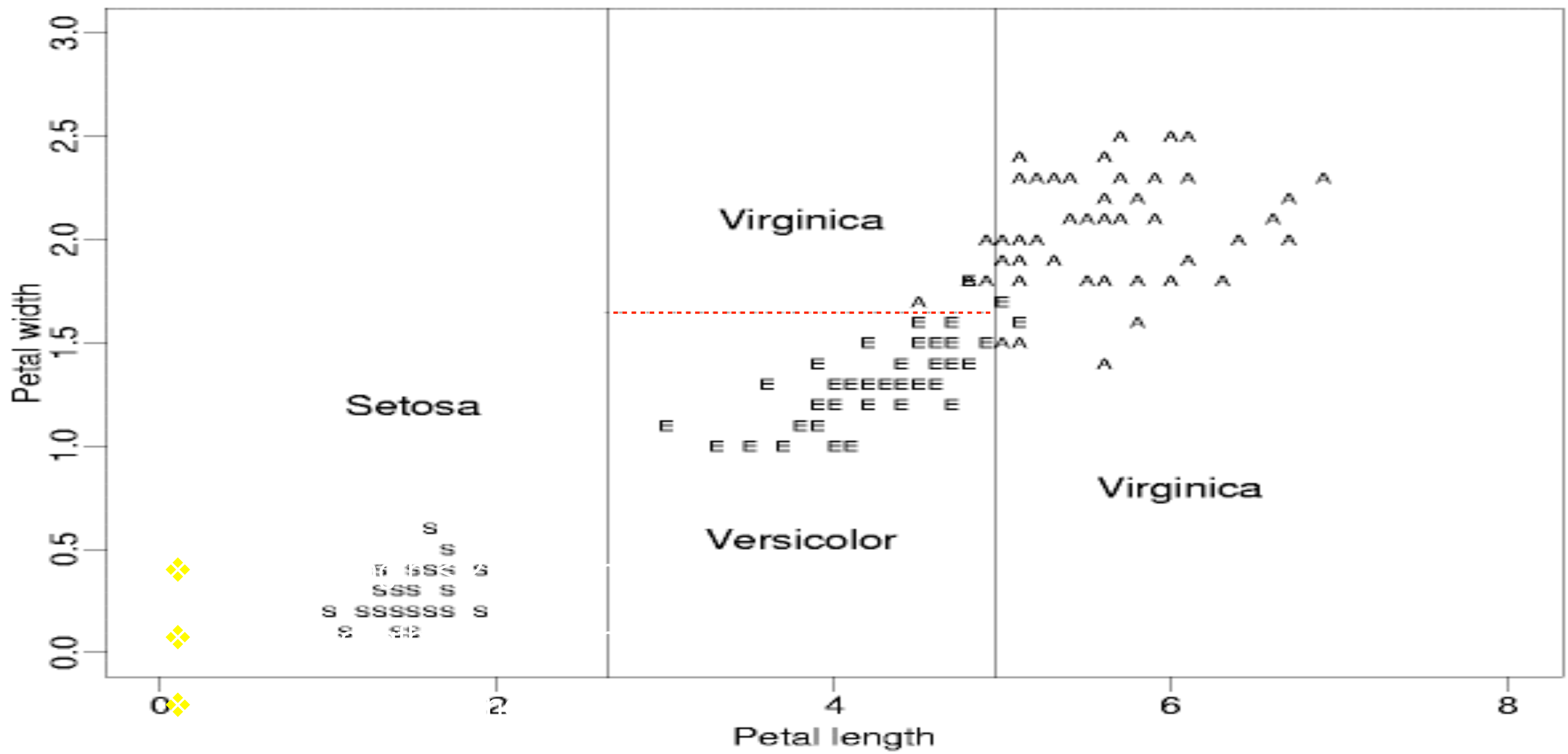
- Datapoints: 150 (3 balanced classes, from UC Irvine ML Depository)
- Collected by Edgar Anderson (two species collected from Gaspé, Quebec!)
- Features: Petal Width, Petal Length, Sepal Width, and Sepal Length

Iris Dataset: Scatterplot

(https://commons.wikimedia.org/wiki/File:Iris_dataset_scatterplot.svg)

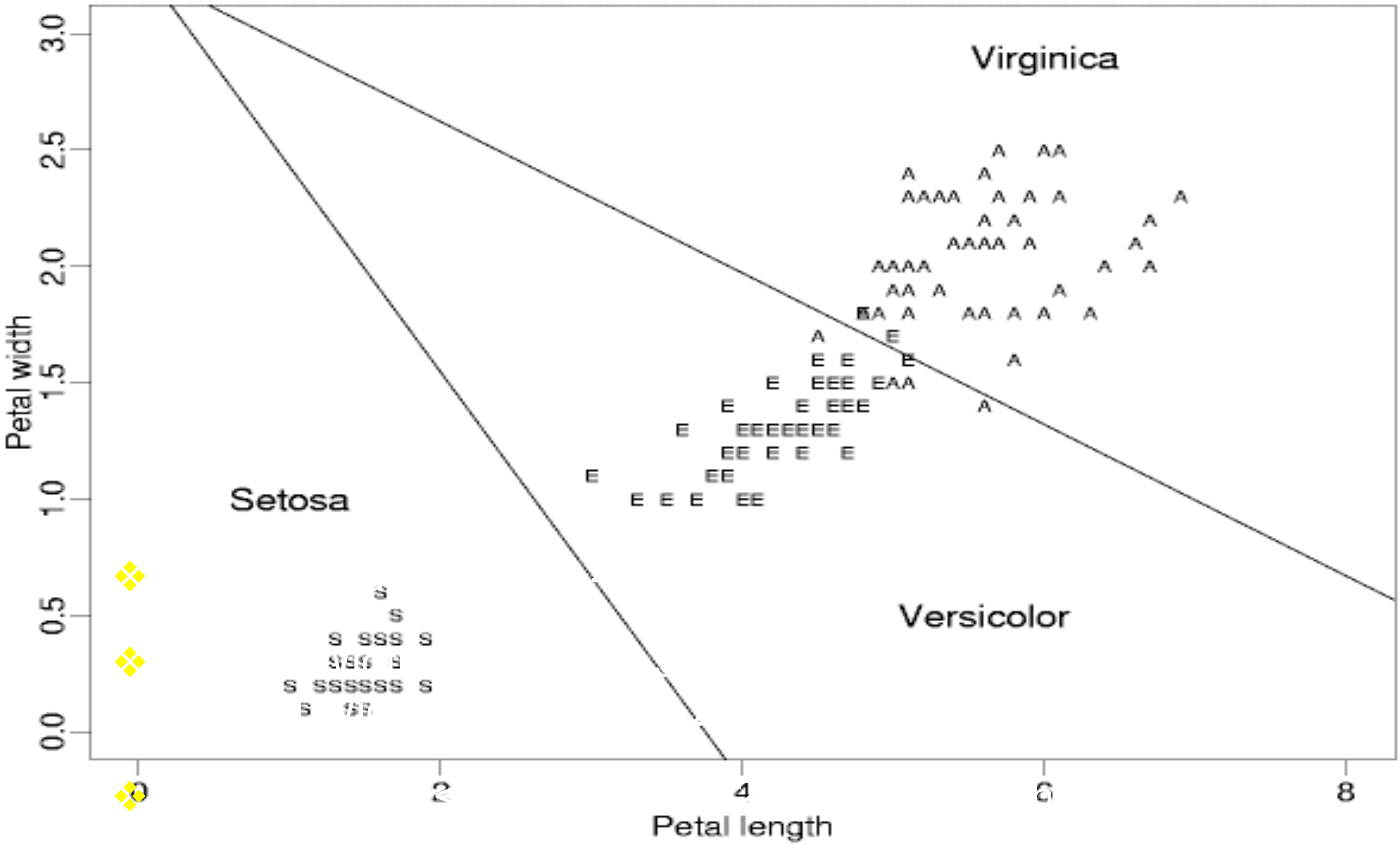


Decision Tree



- if Petal Width < 1.65 then *Versicolor*
- if Petal Width >= 1.65 then *Virginica*

Fisher's Linear Discriminant (1936)



Summary of Machine Learning

- Classical Machine Learning → learns mappings
- Deep Learning → learns representations
- LLMs → learn general models usable across tasks