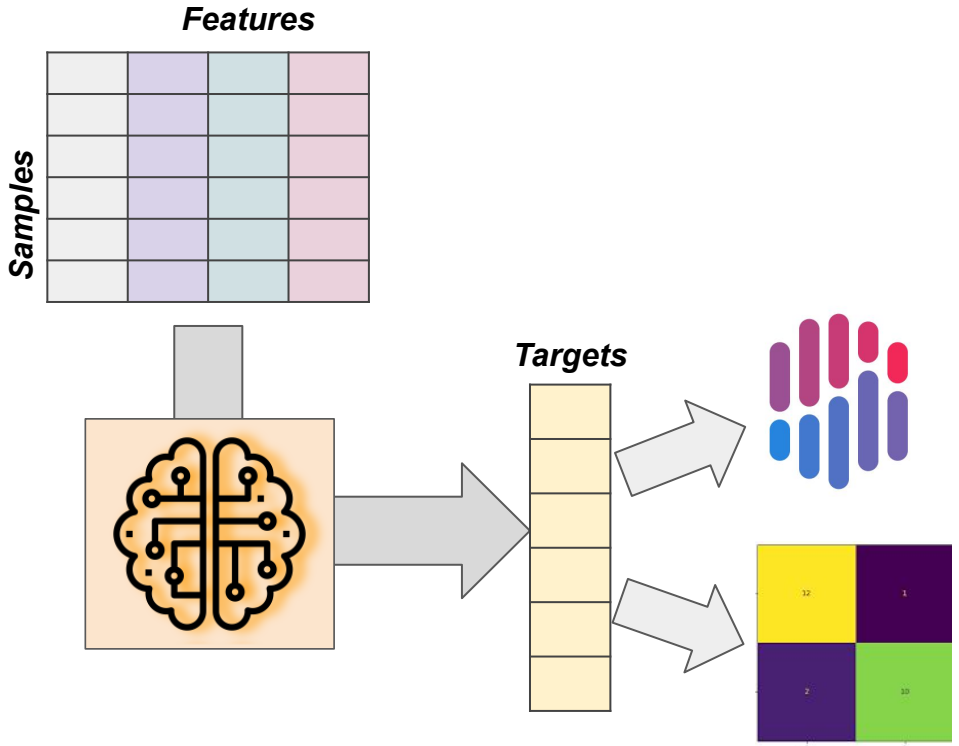
Introduction to Machine Learning (ML) for Tabular Data

Prof. Ishaan Gupta

Functional Genomics Lab
Department of Biochemical Engineering and Biotechnology
Indian Institute of Technology, Delhi

Machine Learning for Tabular Data

- **Data Landscape** in Biology/Biomedicine
- Introduction to **Tabular Data**
- **Machine Learning** Workflow
- **Challenges** in Handling Tabular Data
  - Data Cleaning
  - Handling Class Imbalance
  - Feature Selection
- **Models**
  - Linear Regression and SVM
  - Random Forest and Gradient Boosting
  - Deep Learning Models
- Model **Training**
- Model **Evaluation** and **Interpretability**
- Model **Deployment**
- Translational **Application**



Data Landscape in Biology: A Machine Learning Playground

- Modern biology/biomedical research generates **vast**, **complex**,**heterogeneous** and **high-dimensional** data.
- Some of the common types of biomedical data are as follows:

Data Type	Example	ML Models
Images	Histopathology slides, MRI, CT scans	CNN, ViT, YOLO, Detectron
Sequences	DNA, RNA, Protein sequences	RNN, LLM
Tabular Data	Clinical lab tests, omics profiles, Electronic Health Records	LR, RF, SVM, XGBoost

- Most clinical and experimental datasets are represented in tabular form.
- Examples:
  - **Gene Expression Matrices:** Patients × Gene expression levels
  - **Proteomic Profiles:** Protein abundances
  - **Patient Metadata:** Age, BMI, comorbidities, treatment outcomes
  - **Electronic Health Records (EHRs):** Integrated clinical variables across populations

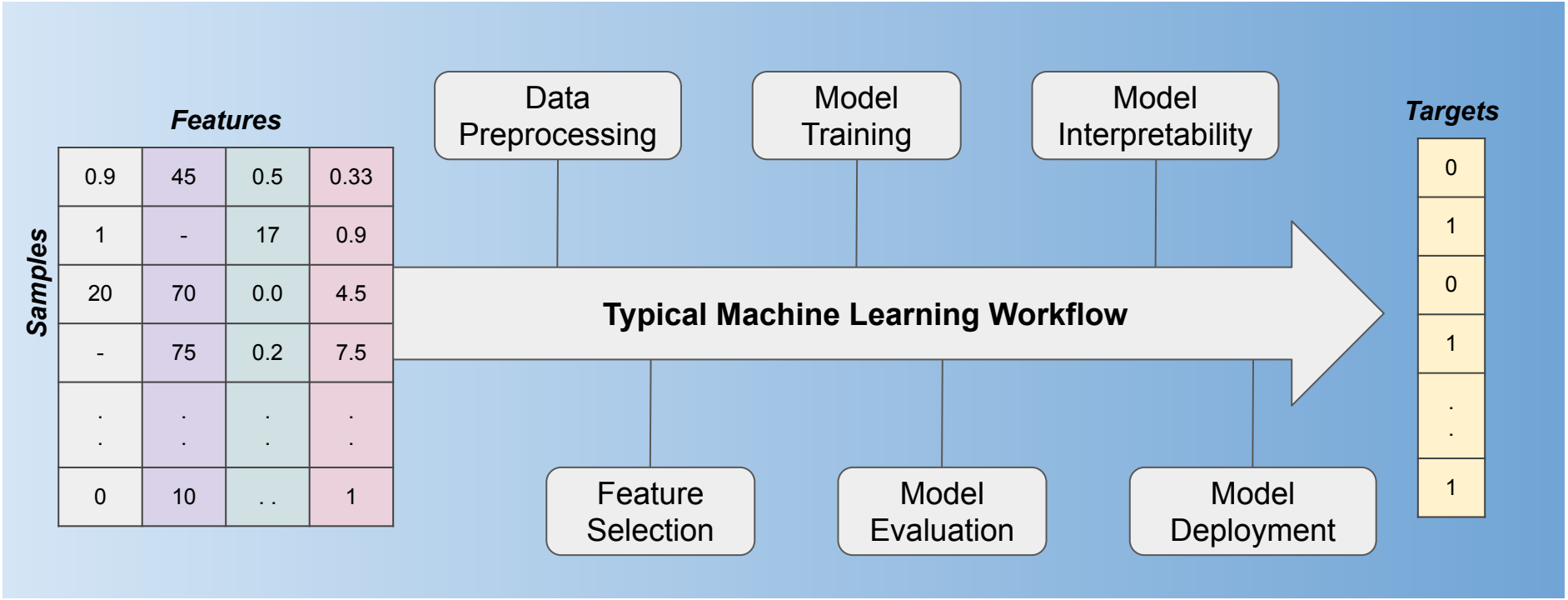
What is Tabular Data?

- Tabular data are often represented in a matrix-like format — rows and columns.
- **Rows** correspond to **samples, patients, or observations**.
- **Columns** represent **features, variables, or biomarkers**.
- Key challenges in working with tabular data are as follows:
  - **Heterogeneous data**
  - **Data sparsity**
  - **High dimensionality (m << n).**
  - **Correlated variables**
  - **Imbalanced classes**
  - **Noisy data**
  - **Outliers**

Sample	*V1*	*V2*	*. . .*	*Vn*	*Target*
S1	0.9	45	. . .	0.5	0
S2	1	-	. . .	17	1
S3	20	70	. . .	0.0	0
S4	-	40	. . .	0.2	1
. . .	. . .	. . .	. . .	. . .	. . .
Sm	0	10	. . .	0.9	1

*Tabular data consisting of **m** samples and **n** variables with a **binary target** (0 & 1).*

The Machine Learning Workflow

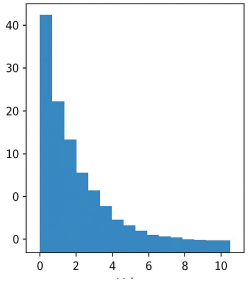


Data Preprocessing

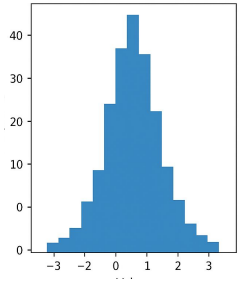
- **Data Cleaning**: remove duplicates, *imputing missing values.
- **Feature Encoding**: converting categorical variable (e.g. gender) into numeric.
- **Normalization/Scaling** (*z-score, min-max): Ensures equal contribution of all features during model training
- **Handling Outliers**: identifying and removing outliers
- **Feature Selection**: removing multicollinearity (*ANOVA) and dimensionality reduction
- **Over/Under Sampline**: handling class imbalance
- **Feature Engineering**: Introducing meaningful features derived from existing features.
    - Example: In tumor classification, tumor volume = $0.5xL \times W^2$ is more meaningful than tumor length or width alone.
- **Splitting the Dataset**:
    - Data split: Training (70%), Validation (15%), Test (15%)

Data Preprocessing *(Continued)*

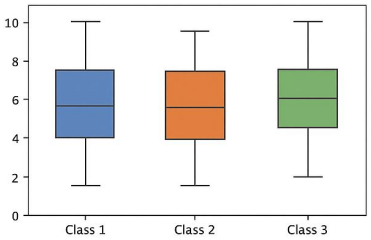
- **Imputation**
  - **Lower detection limit (LOD)**.
  - **Mean** or **Median**
  - Imputation Algorithms: **KNN**
- **Z-Score Normalization**
  - Mean = 0
  - Standard deviation = 1
- **Feature Selection using ANOVA**
  - Identify features that significantly (**F-statistics**) differ across classes i.e important features otherwise collinear/redundant features.

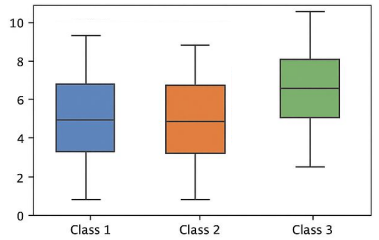


$$z = \frac{x - \mu}{\sigma}$$



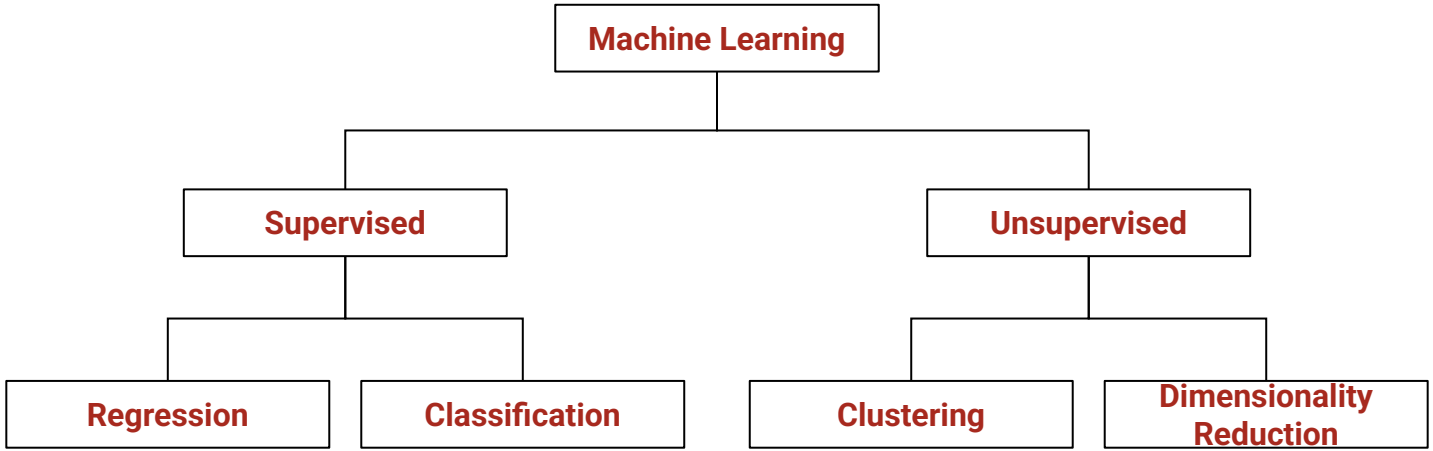
$$F = \frac{\text{Variance between groups}}{\text{Variance within groups}}$$





Machine Learning

- Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables computers to **learn from data and make predictions without being explicitly programmed**.
- Can be broadly divided into **Supervised** and **Unsupervised** learning, based on the availability of labeled data during training.
- Supervised learning, the most common ML type, is divided into **Regression** (continuous target) and **Classification** (categorical target).



Objective of a Machine Learning Model

Let's suppose we have a model (f) with parameters $\beta_1, \beta_2, \ldots, \beta_n$ and b.

$$f(x) = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + b$$

where $x = x_1, x_2, \ldots x_n$ represent rows of feature matrix

Sample	x1	x2	. . .	xn	Target
S1	0.9	45	. . .	0.5	0

Our objective is to find the optimal parameters:

$$\theta = \{\beta_1, \beta_2, \ldots, \beta_n, b\}$$

such that loss (prediction error) is minimized:

$$min \, // y - f(x) //^2$$

Linear Models: Linear Regression (LR) and SVM

Linear Regression (LR)

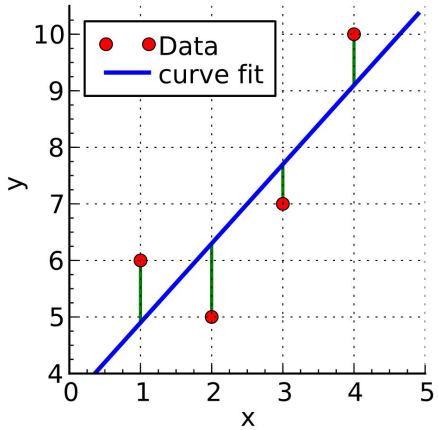


Image source: Wikipedia – "Linear regression"

- Fits a **best-fit line** (**blue**) through the data capture the relationship between input features and the target variable.
- Goal: Minimize the difference between predicted and actual values (loss).

Support Vector Machine (SVM)

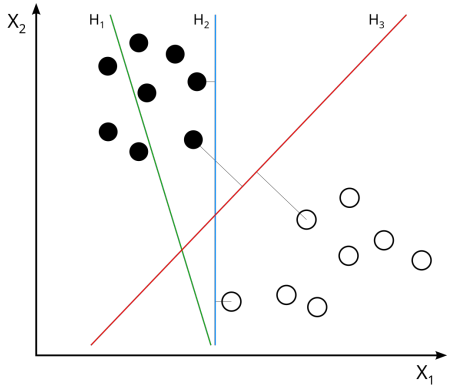


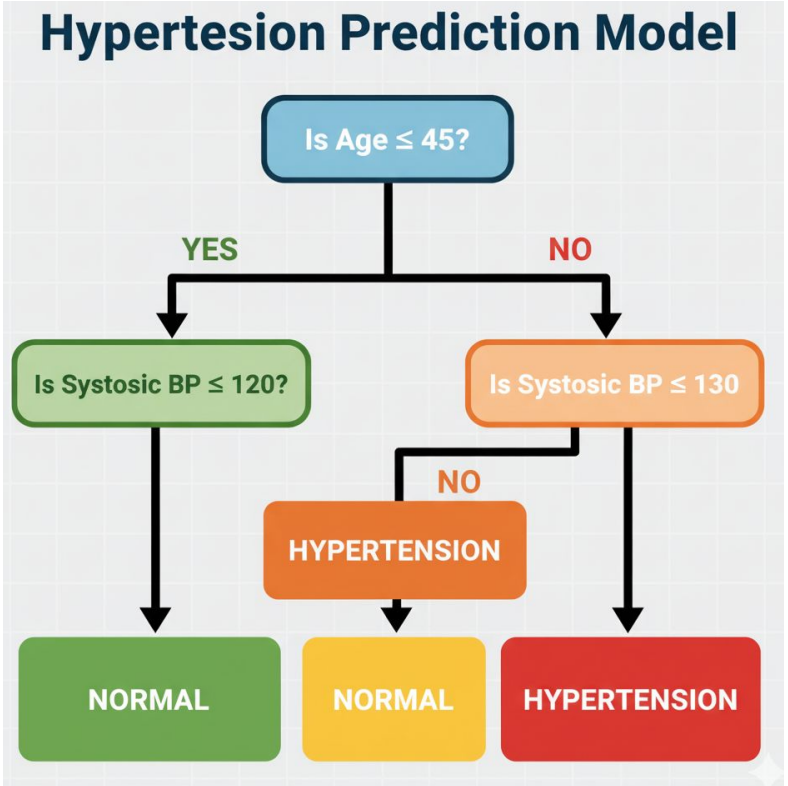
Image source: Wikipedia – "Support vector machine"

- Find the **optimal hyperplane** (**H3: red**) that separates classes.
- Maximize the margin i.e. the distance between the hyperplane and the nearest data points (support vectors).
- Goal: Achieve maximum separation between classes while minimizing misclassification.

 # The model shown earlier is a multiple linear regression model.

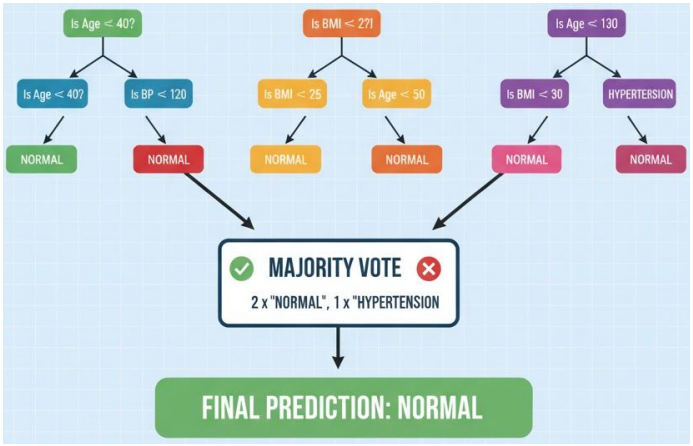
Tree Based Model: Decision Tree

- A decision tree learns simple **if–then rules** from data to make predictions.
- Works by splitting data into branches based on feature values, forming a tree-like structure.
  - **Internal node** :decision based on a features
  - **Branch:** outcome
  - **Leafs:** final prediction
- Goal: Create the **most homogeneous possible subsets at each split** using metrics like Gini impurity, Entropy, or Variance reduction.
- **Ensembles** of decision trees form models like **Random Forest** and **Gradient Boosting** for better accuracy and robustness.



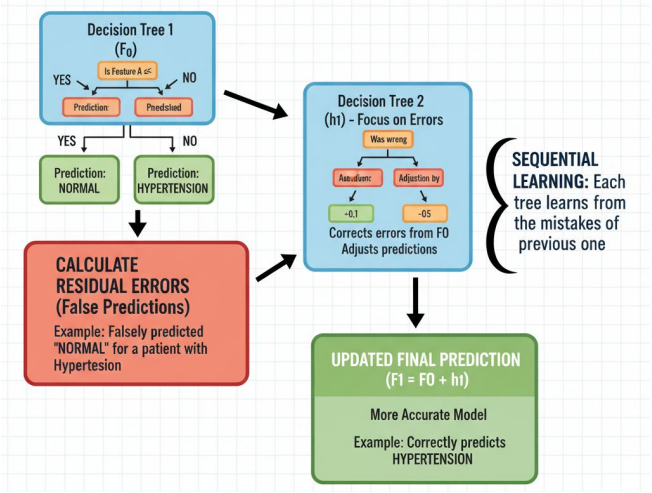
Ensemble Models: Random Forests and Gradient Boosting

Random Forest (RF)



- An **ensemble of many decision trees**, each trained on random samples and features.
- Final prediction = majority vote (classification) or average (regression).

Gradient Boosting (GB)



- Builds trees sequentially, each **new tree corrects the errors of the previous ones**.
- Learns by minimizing a loss function step by step.

Deep Learning Models: Tabular foundation models (TFMs)

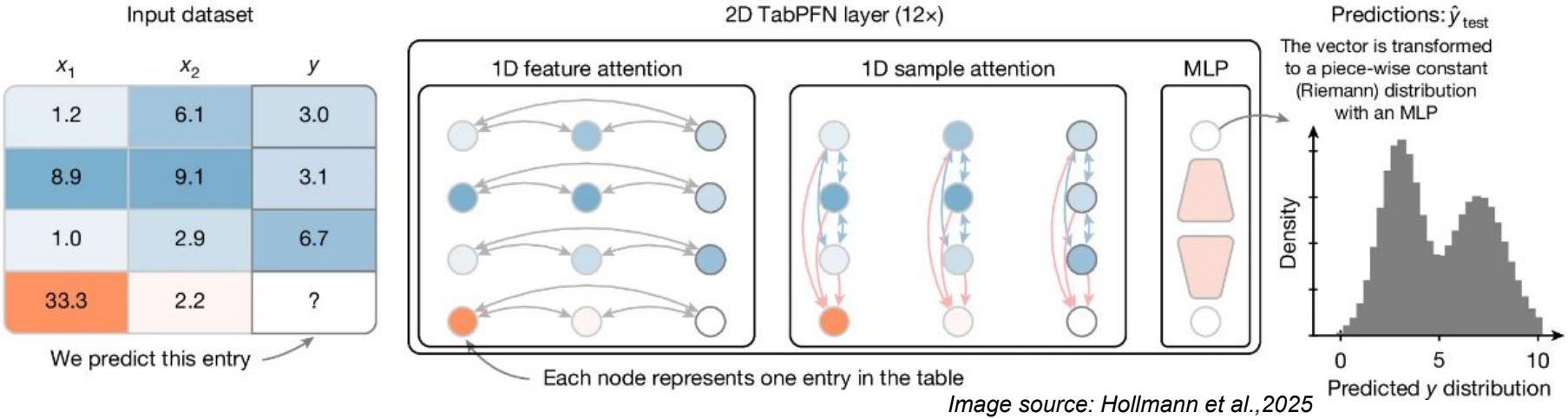


Image source: Hollmann et al.,2025

- **Traditional deep learning** models requires large datasets for training, but experimental tabular data often has m << n.
- TFMs overcome this limitation by **learning general tabular patterns** during pre-training using **millions of synthetic tabular datasets**.
- A pre-trained TFM can achieve **high performance even with few training samples**.

Tabular foundation models: TabPFN

- TabPFN stands for **Tabular Prior-Data Fitted Network.**
- It's a pre-trained **Transformer Encoder** model. Trained on 130 million synthetic datasets.
- Training objective was to **predict the masked target values** (y) in those tabular datasets.
- It generally works well on small datasets (**<10,000 samples**) in a single forward pass.

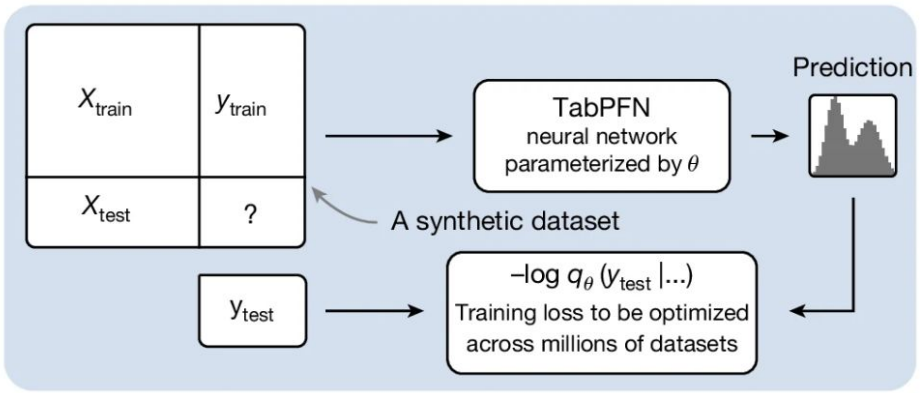


Image source: Hollmann et al.,2025

- TabPFN processes the entire dataset (features and target labels) as **single set of tokens.**
- **No need of data preprocessing** (imputation, class imbalance, data cleaning, etc.)

Tabular foundation models: TabPFN

- Benchmarked on **OpenML CC-18** dataset (collection of 72 real-world classification datasets) and from **published studies**.
- During evaluation phase the target variable can be continuous (regression) or categorical variable (classification)
- Significantly **outperformed** the ML (XGBoost, LightGBM, & CatBoost) and DL models.
- Performance drops on

  larger datasets (>10,000 samples).

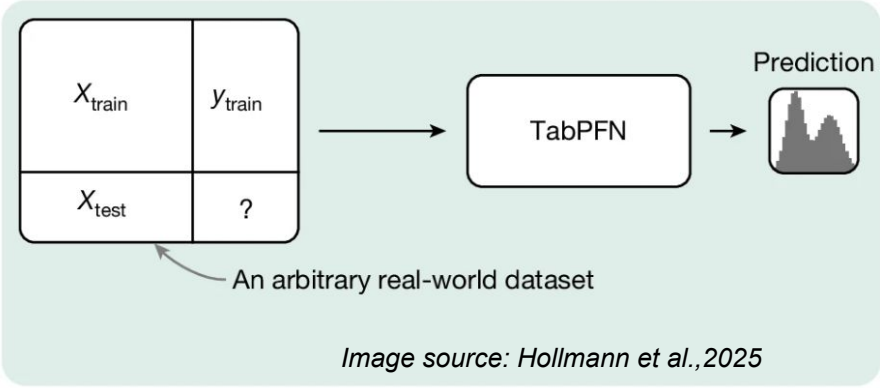
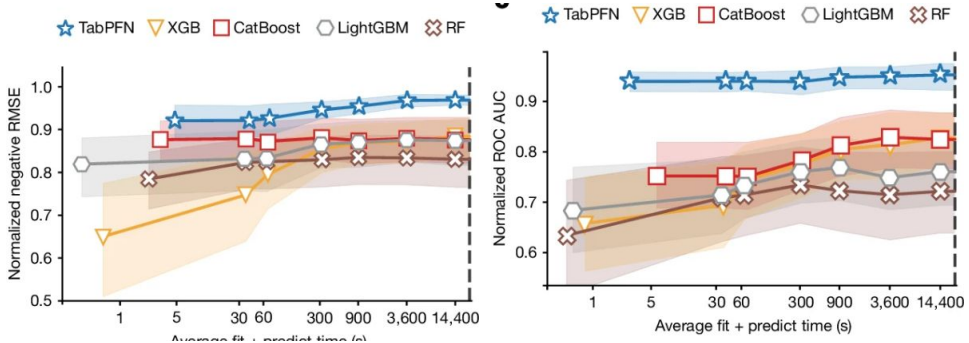


Image source: Hollmann et al.,2025



Model Evaluation

- During the evaluation phase, the trained model is tested on unseen dataset (**test set**) to assess its **accuracy**, **generalization**, and **reliability** using various **evaluation metrics**.

True Positive	*False Positive*
False Negative	*True Negative*

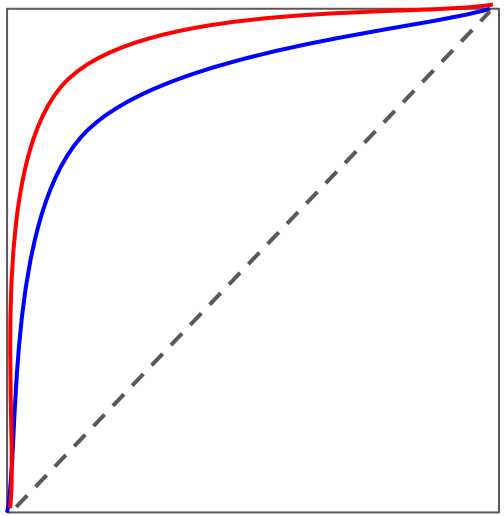
Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

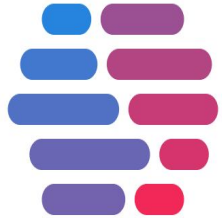
$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$



AUROC Curve

Model Interpretability

- **Self-Explainable Models** (**Linear Regression**, Decision Trees, Random Forests)
  - In case of Linear Regression, **model's parameters** (β_i) indicates the effect of each feature on model's predictions.
  - Example: A model for BP prediction, with age ($\beta = 0.6$) as a feature. The β indicates that for every 1-year increase in age, BP increases by 0.6 mmHg, holding other features constant.
- **SHAP** (SHapley Additive exPlanations)
  - Based on **game theory** (Shapley values).
  - Explains each prediction by fairly **distributing contribution among features.**
- **LIME** (Local Interpretable Model-Agnostic Explanations)
  - Explains a **single prediction** by **approximating the complex model with a simple one**.
  - Perturbs input slightly to see how predictions change



Model Deployment

- Model deployment is the **final stage** of a machine learning workflow, where the trained model is made available for **silent trails** and **real-world use**.
- **Development** (Jupyter) to **Production** (Web App)
- Typical steps involves: Saving model (.pkl) →Build an App (Streamlit) → Host model (Local Machine) → Monitor real world performance
- Some of the lightweight and easy-to-deploy frameworks for model deployment and visualization are as follows:

Streamlit

Shiny for Python

Plotly Dash

Translational Applications (Case Studies)

- **Risk Stratification** (High vs Low) Model for pediatric acute myeloid leukemia (**AML**) and acute lymphoblastic leukemia (**ALL**) *[Al-Hussaini et al, 2024]*

- ML based **biomarker discovery** for **pre-metabolic syndrome** and **metabolic syndrome** using lipidomics profiling *[Huang et al, 2024].*

- **Precision Medicine** for **Multiple Myeloma** Patients Using ML-Enabled Proteomic Profiles *[Katsenou et al., 2023].*

- **Disease prediction** in **Incident Atrial Fibrillation** Using Electronic Health Record Data with Machine Learning *[Tiwari et al., 2019].*

Conclusion & Key Takeaways

- Most biomedical and clinical datasets are tabular (heterogeneous, noisy, and often imbalanced).
- Preprocessing and feature selection form the foundation for reliable modeling.
- Gradient Boosting models remain strong baselines for tabular data.
- Tabular Foundation Models (TFMs) are emerging, enabling accurate predictions even on small datasets.
- A rigorous evaluation strategy is crucial to build a robust model for real world application
- Interpretable ML models transform data into actionable biomedical insights.
- Integrating ML with EHR, proteomic, and genomic data accelerates precision medicine through data-driven insights.

Thank You

GitHub Link

https://github.com/rajanbit/ML-for-Tabular-Data-ICGEB-Workshop-2025

OR

> github.com

> Search “icgeb workshop 2025”

> Open the search result