Supervised Learning Models

Linear Regression

Use: Predict continuous target with linear relationship.
Strengths: Simple, interpretable, efficient.
Weaknesses: Fails on non-linear patterns, sensitive to outliers and collinearity.
Typical Performance: Baseline—e.g., mean squared error (MSE) on housing data ≈10–15% above tree models.

Logistic Regression

Use: Binary classification, outputs class probability.
Strengths: Fast, interpretable; with L2 or elastic-net works well in high-dimensional cases.
Weaknesses: Limited to linear decision boundaries.
Benchmark Notes: On small/structured data, Logistic with elastic-net slightly lags behind linear SVM; performance depends heavily on regularization.

Decision Tree (Classifier/Regressor)

Use: Non-linear splits on structured/tabular data.
Strengths: Interpretable, handles categorical data.
Weaknesses: Prone to overfitting; requires tuning (depth, min_samples).
Typical Performance: Baseline; Random Forest greatly improves accuracy.

Random Forest (Classifier/Regressor)

Use: Ensemble of many decision trees via bagging.
Strengths: High accuracy, robust to overfitting, minimal tuning.
Weaknesses: Slower to predict large ensembles; less interpretable.
Benchmark Notes: Often second-best after LightGBM; top performer across many tabular tasks.

Gradient Boosting (e.g., GBM, XGBoost, LightGBM)

Use: Boosted decision trees.
Strengths: Top accuracy on tabular data; handles non-linearity and interactively.
Weaknesses: Sensitive to hyperparameters, slower training (especially XGBoost).
Benchmark Notes: XGBoost/CatBoost/LightGBM dominate tabular benchmarks, outperform deep models.

Support Vector Machine (Classifier/Regressor)

Use: Max-margin classification; SVR does regression.
Strengths: Works well on small to medium data; kernels allow non-linearity.
Weaknesses: Doesn't scale well; hard to tune C, kernel.
Benchmark Notes: Linear SVC often beats Logistic Regression; performs best on diseases like heart/diabetes; widely use.

K-Nearest Neighbors (Classifier/Regressor)

Use: Distance-based classification/regression.
Strengths: Simple, non-parametric; effective when class boundary is complex.
Weaknesses: Slow at inference; sensitive to scaling/noise; need to tune k.
Benchmark Notes: Good baseline; on MNIST, yields ≈0.96% error.

Naive Bayes (Gaussian)

Use: Probabilistic classification under feature independence.
Strengths: Fast, robust, works with small data.
Weaknesses: Independence assumption often false but still performs well in practice.
Benchmark Notes: Second-most used model; often wins in disease prediction tasks.

Multilayer Perceptron (Classifier/Regressor)

Use: Feedforward neural network for non-linear mappings.
Strengths: Capable of capturing complex relationships.
Weaknesses: Needs careful tuning; slower; risk of overfitting.
Benchmark Notes: On Wisconsin Breast Cancer dataset, MLP reaches ~99% accuracy. On MNIST, 2-layer MLP ~1.6% error, deep MLP ~0.35%.

Unsupervised Learning Models

K-Means Clustering

Use: Partition data into k groups by minimizing within-cluster variance.
Strengths: Fast and simple.
Weaknesses: Assumes spherical clusters of equal size; needs k known.
Evaluation Metrics: Silhouette score, Adjusted Rand Index (ARI), NMI, Fowlkes–Mallows.

DBSCAN

Use: Density-based clustering; identifies core, reachable, and noise points.
Strengths: Detects arbitrary shapes, handles outliers.
Weaknesses: Hard to tune epsilon/minPts; struggles with varying densities.

Agglomerative Clustering

Use: Hierarchical clustering via merging bottom-up.
Strengths: Doesn't need a fixed number of clusters; produces dendrogram.
Weaknesses: Computationally expensive; choice of linkage matters.

Gaussian Mixture Model (GMM)

Use: Clustering with soft probabilistic assignments.
Strengths: Captures cluster shape through covariance.
Weaknesses: Requires correct number of components; can converge poorly.

PCA

Use: Linear dimensionality reduction.
Strengths: Captures directions of maximal variance, speeds up learning.
Weaknesses: Only captures linear structure.

t-SNE & UMAP

Use: Non-linear dimensionality reduction for visualization.
Strengths: Great for visual data exploration.
Weaknesses: Not reproducible exactly; insensitive to clustering evaluation.

Autoencoder

Use: Learn compressed representation; unsupervised feature learning.
Strengths: Non-linear reduction; customizable.
Weaknesses: Requires more tuning and compute.

Self-Organizing Map (SOM)

Use: Grid-based neural clustering and visualization.
Strengths: Good for exploratory mapping and spatial relationships.
Weaknesses: Hard to scale; manual architecture/design.

Comparative Performance

- **Tabular Supervised Tasks:**
    - Gradient boosting (XGBoost, LightGBM, CatBoost) consistently top performers.
    - Random Forest: strong and robust; easier tuning.
    - SVM and MLP shine in certain domains like disease prediction or when proper tuning is applied.

- **Neural Deep Models vs. Traditional ML:**
    - On structured/tabular data, deep learning rarely outperforms GBMs. Only on very large and complex tabular sets do they occasionally win.
- **Unsupervised Clustering:**
    - Performance varies; metrics like ARI, NMI are used to evaluate with ground truth.
    - Spectral clustering sometimes outperforms K-Means (e.g., ARI 0.982 vs 0.827).

✅ Choosing the Right Model

Scenario	Recommended Models
Tabular structured data	GBMs (LightGBM/XGBoost), then Random Forest, SVM, MLP.
Interpretability needed	Logistic Regression, Decision Tree.
Small dataset	SVM, Logistic, Naive Bayes, KNN.
Complex non-linearity	MLP or tree-based models.
Unsupervised / visualizing	PCA/Umap/t-SNE, K-Means, DBSCAN/GMM depending on shape/density.

Evaluation metrics:

- Classification: accuracy, precision, recall, F1, AUC (beware of accuracy paradox on imbalanced data).
- Regression: RMSE, MAE, R^2.
- Clustering: Silhouette, ARI/NMI (needs ground truth), Fowlkes–Mallows index.

🧠 Real-World Examples

- **MNIST (handwritten digits):**
    - KNN ~0.96% error; SVM ~0.56%; deep CNN ensembles ~0.09%.

- **Wisconsin Breast Cancer:**
    - MLP reaches ~99% accuracy. Other models (SVM, Logistic, NN) >90%.
- **Disease Prediction Tasks:**
    - Across 49 diseases, SVM most-used; RF highest accuracy ~53% of the time; SVM won ~41%.

🧩 Summary & Best Practices

1. **Start simple**: Logistic/Linear or Decision Tree as baseline.
2. **Use ensemble/boosting**: LightGBM or Random Forest for most tabular tasks.
3. **Scale and tune**: SVM/MLP if dataset is small and you can tune well.
4. **Evaluate properly**: Use cross-validation and relevant metrics (e.g., F1 for imbalance).
5. **Validate results**: Compare across multiple models on held-out benchmark datasets or external validation