

Week 0b: Supervised Learning

The Prediction Challenge

Machine Learning for Smarter Innovation

BSc Innovation & Design Thinking

- 1 Part 1: The Challenge
- 2 Part 2: Linear + Regularization
- 3 Part 3: Nonlinear Methods
- 4 Part 4: Synthesis

1. Real Estate Price Prediction

The Business Problem

- Predict house prices from features
- Features: size, bedrooms, location, age
- Target: price in thousands
- Training data: 10,000 historical sales

Sample Data Points

Size	Beds	Age	Price
1200	2	5	250k
2500	4	10	450k
1800	3	2	380k

charts/real_estate_scatter.pdf

Supervised learning transforms labeled examples into predictive models - regression predicts continuous values from multiple correlated features

2. Linear Regression as Baseline

Mathematical Foundation

- Model: $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \epsilon$
- Where y = price, x_i = features
- Goal: Find best-fitting line/plane
- Method: Minimize squared errors

Assumptions

- Linear relationship
- Independent features
- Constant variance
- Normal errors

charts/linear_regression_fit.pdf

Linear regression assumes additive relationships - optimal under Gauss-Markov conditions but restrictive for real world complexity

3. Classification vs Regression

Regression Problems

- Predict continuous values
- Examples: price, temperature, stock return
- Output: Real numbers
- Metrics: MSE, MAE, R-squared

Classification Problems

- Predict discrete categories
- Examples: spam/ham, buy/sell/hold
- Output: Class labels
- Metrics: Accuracy, precision, recall

charts/regression_vs_classification.pdf

Problem formulation determines algorithm choice - continuous targets require regression, discrete categories need classification with appropriate loss functions

4. The Curse of Dimensionality

Feature Explosion Problem

- Real estate: 20+ features
- Interactions: $2^{20} = 1,048,576$ combinations
- Sample: 10,000 data points
- Ratio: 104 interactions per data point

Mathematical Challenge

- High-dimensional space is mostly empty
- Distance metrics become meaningless
- Overfitting becomes inevitable
- “Hughes phenomenon” (Hughes 1968): Performance degrades as features increase beyond sample capacity

charts/curse_dimensionality.pdf

Curse of dimensionality: volume grows exponentially while data remains sparse - distances become uninformative and nearest neighbors lose meaning (Bellman 1961)

5. Feature Interactions Explode Combinatorially

Combinatorial Mathematics

- Linear terms: n features
- Pairwise: $\binom{n}{2} = \frac{n(n-1)}{2}$
- Three-way: $\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$
- All subsets: $2^n - 1$

Real Estate Example (n=20)

- Linear: 20 terms
- Pairwise: 190 interactions
- Three-way: 1,140 interactions
- Total possible: 1,048,575 terms

charts/feature_combinations.pdf

Combinatorial explosion creates 2^n possible interactions - regularization and feature selection become necessary to prevent overfitting in high dimensions

6. OLS with Worked Example

Ordinary Least Squares

- Minimize: $\sum_{i=1}^n (y_i - \hat{y}_i)^2$
- Solution: $\beta = (X^T X)^{-1} X^T y$
- Assumptions: $X^T X$ is invertible
- Unbiased estimator under Gauss-Markov

Worked Example Data: (1, 2, 3) predicts (2, 4, 6)

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}, y = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \quad (1)$$

$$\beta = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad (2)$$

charts/ols_example.pdf

OLS provides minimum variance unbiased estimators under Gauss-Markov assumptions - closed-form solution enables fast computation for linear models

7. Ridge and LASSO Regularization

Ridge Regression (L2)

- Minimize: $\|y - X\beta\|^2 + \lambda\|\beta\|^2$
- Shrinks coefficients toward zero
- Keeps all features
- Solution: $\beta = (X^T X + \lambda I)^{-1} X^T y$

LASSO Regression (L1)

- Minimize: $\|y - X\beta\|^2 + \lambda\|\beta\|_1$
- Sets some coefficients to exactly zero
- Automatic feature selection
- No closed-form solution

Elastic Net

- Combines L1 + L2: $\alpha\|\beta\|_1 + (1 - \alpha)\|\beta\|^2$
- Balances selection and shrinkage

charts/ridge_lasso_comparison.pdf

Regularization adds penalty terms to loss function - L2 shrinks coefficients, L1 enforces sparsity, Elastic Net combines both strategies

8. SUCCESS: Perfect on Linearly Separable Data

Linear Model Triumphs

- Iris setosa classification: 100% accuracy
- House price in suburbs: $R^2 = 0.94$
- Linear trend prediction: $MSE = 0.01$
- Feature importance: Interpretable

Why It Works

- Underlying relationship is linear
- Features are independent
- Low noise in measurements
- Sufficient training data

charts/linear_success_cases.pdf

9. FAILURE: Terrible on XOR, Nonlinear Boundaries

Linear Model Failures

Dataset	Linear	Tree
XOR	50%	100%
Circles	52%	98%
Moons	58%	94%

Why XOR Fails:

No single line separates (0,0), (1,1) from (0,1), (1,0)

charts/linear_failure_cases.pdf

10. Root Cause: Linear Assumption Too Restrictive

Mathematical Limitation

Linear model $y = w^T x + b$ creates hyperplane boundaries that cannot curve, bend, or create islands

Real-World Implications

Customer behavior, stock patterns, medical diagnosis, image recognition all exhibit nonlinear relationships

charts/linear_vs_nonlinear_boundaries.pdf

11. Regularization Tradeoff

Bias-Variance Tradeoff

$\lambda = 0$: high variance; $\lambda \rightarrow \infty$: high bias; optimal λ found via cross-validation

Practical Guidelines

Ridge for stability, LASSO for selection, Elastic Net combines both; use grid search for λ

charts/regularization_tradeoff.pdf

12. Human Introspection: How YOU Divide Decision Space

Your Natural Decision Process

- “Is income \geq 50k?” - \rightarrow Split population
- “If yes, is age \geq 40?” - \rightarrow Further split
- “If no, is education \geq 12 years?” - \rightarrow Alternative path
- Continue until clear decision

Hierarchical Thinking

- Start with most important feature
- Recursively subdivide space
- Each split reduces uncertainty
- Stop when confident

charts/human_decision_process.pdf

13. Hypothesis: Trees, Kernels, Ensembles

Decision Trees

- Recursive binary splits
- Non-parametric method
- Handles interactions naturally
- Interpretable rules

Kernel Methods

- Map to higher dimensions
- “Kernel trick” for efficiency
- SVM with RBF, polynomial kernels
- Implicit feature expansion

Ensemble Methods

- Combine multiple weak learners
- Random Forest, Gradient Boosting
- Reduce overfitting through averaging

charts/nonlinear_methods_overview.pdf

14. Zero-Jargon: “20 Questions Game” for Trees

The Game Analogy

- You think of an animal
- I ask yes/no questions
- “Is it bigger than a cat?”
- “Does it live in water?”
- “Is it a mammal?”

Decision Tree Mapping

- Animal = Data point
- Questions = Feature splits
- Final guess = Prediction
- Good questions = Informative features

charts/twenty_questions_tree.pdf

15. Geometric Intuition: Decision Boundaries

Boundary Types

Linear: planes; Trees: axis-aligned rectangles; SVM RBF: curves; Neural nets: arbitrary shapes

Complexity Hierarchy

Linear (most restrictive) → Trees → Kernels → Deep networks (most flexible)

charts/decision_boundaries_comparison.pdf

16. CART Algorithm with Actual Splits

CART Algorithm Steps

- 1 Calculate impurity for current node
- 2 Try all possible splits
- 3 Choose split with highest information gain
- 4 Recurse on child nodes
- 5 Stop when stopping criterion met

Gini Impurity Formula

$$G = 1 - \sum_{i=1}^C p_i^2$$

where p_i is probability of class i

Information Gain

$$IG = G_{parent} - \sum \frac{n_{child}}{n_{parent}} G_{child}$$

charts/cart_algorithm_steps.pdf

17. Full Walkthrough: Build Tree with Numbers

Best Split: Income \geq 55k

Left ($<$ 55k): 1Y, 2N $\rightarrow G_L = 0.444$

Right (\geq 55k): 3Y, 0N $\rightarrow G_R = 0$

Info Gain: $IG = 0.222$

Dataset: Loan Approval

Income	Age	Approved
30k	25	No
60k	35	Yes
40k	45	No
80k	30	Yes
50k	50	Yes
70k	25	Yes

Root Gini: 4 Yes, 2 No $\rightarrow G = 0.444$

charts/tree_building_example.pdf

18. Visualization: Decision Boundaries on 2D Data

Tree Partitioning Process

- Split 1: $x_1 \leq 0.5$ (vertical line)
- Split 2: $x_2 \leq 0.3$ (horizontal line)
- Split 3: $x_1 \leq 0.8$ (vertical line)
- Result: Rectangular regions

Boundary Characteristics

- Always axis-aligned
- Creates rectangular partitions
- Can approximate any boundary
- With enough splits

charts/tree_2d_boundaries.pdf

19. Why It Works: Piecewise Approximation

Universal Approximation

- Any function can be approximated
- By piecewise constant functions
- With sufficient partitions
- Trees implement this naturally

Mathematical Foundation

- Step functions are dense in L^2
- Trees create step functions
- More splits = better approximation
- Regularization prevents overfitting

charts/piecewise_approximation.pdf

Benchmark Results

Dataset	Linear	Tree	SVM
Iris	96%	96%	98%
Wine	94%	90%	96%
Digits	92%	85%	98%
Breast Cancer	95%	93%	97%
XOR	50%	100%	100%
Circles	52%	98%	100%

Key Insights

- Linear: Good on linear data
- Trees: Excel on discrete features
- SVM: Best overall performance
- No universal winner

charts/algorithm_comparison_table.pdf

21. Implementation: sklearn Ensemble Methods

Random Forest

- Bootstrap sampling of data
- Random subset of features
- Average predictions
- Reduces overfitting

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)
```

Gradient Boosting

- Sequential weak learners
- Each corrects previous errors
- Weighted combination
- Often best performance

charts/ensemble_methods_performance.pdf

Complexity Progression

- **Linear:** $y = w^T x + b$
- **Tree:** Recursive partitioning
- **Ensemble:** Multiple tree combination
- **SVM:** Kernel-based mapping

Computational Complexity

- Linear: $O(nd)$ training
- Single tree: $O(nd \log n)$
- Random forest: $O(tnd \log n)$
- SVM: $O(n^2 d)$ to $O(n^3 d)$

charts/algorithm_landscape.pdf

23. When to Use Each: Interpretability vs Accuracy

Linear Models

- High interpretability, fast
- Use: Regulatory needs, simple patterns

Decision Trees

- Moderate interpretability
- Use: Rule extraction, mixed data

Ensemble Methods

- Highest accuracy, robust
- Use: Performance critical

SVM

- Kernel flexibility
- Use: High dimensions, small data

charts/interpretability_accuracy_tradeoff.pdf

Pitfall 1: Data Leakage

- Using test data during training
- Solution: Strict train/test separation

Pitfall 2: Wrong Metric

- Optimizing accuracy on imbalanced data
- Solution: Use F1, AUC-ROC for imbalanced cases

Pitfall 3: Feature Scaling

- Forgetting to normalize features
- Solution: StandardScaler before distance-based methods

Pitfall 4: Overfitting

- Too complex model for small data
- Solution: Regularization, cross-validation

Pitfall 5: Ignoring Assumptions

- Linear model on nonlinear data
- Solution: Validate assumptions, use nonlinear methods

Pitfall 6: No Baseline

- Not comparing to simple baseline
- Solution: Always start with logistic/linear regression

Most supervised learning failures stem from data leakage, inappropriate metrics, or model mismatch - systematic validation prevents these errors

24. Modern Applications: Production ML Pipelines

Pipeline Stages

Data ingestion → Feature engineering → Training/validation → Deployment
→ Monitoring

Industry Applications

Credit scoring (gradient boosting), recommendations (ensembles), fraud
detection (anomaly), medical diagnosis (interpretable)

charts/production_ml_pipeline.pdf