

Business Data Analysis

MGM509

Lecture 4

Avoiding Pitfalls: Overfitting,
Evaluation, and Complexity Control

Lecture
4

What we will cover

1. What overfitting is and why it happens?
2. How to detect overfitting in models
3. Practical strategies to avoid or reduce overfitting
4. The balance between fitting the data and generalizing to the real world

Lecture
04

Overfitting, Why it Happens, and How to Detect it?



Lecture
4

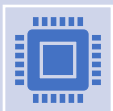
What Is Overfitting?



Overfitting occurs when a model learns patterns specific to the training data, including noise or random fluctuations



These patterns don't generalize well to new, unseen data



The model may appear to perform well during training but fails in real-world applications



In short: an overfit model is too closely tailored to past data to be useful in the future

Lecture
4

Overfitting vs. Generalization

- **Goal of predictive modeling:** find a model that generalizes
- A model that memorizes training data is **not predictive** — it's descriptive
- Generalization is the **true measure of a model's value** .

Generalization	Overfitting
The model performs well on new data	The model performs well on training data only

Lecture
4

Overfitting in Business Contexts

- In business, predictive performance on new data is the priority
- Overfit models can cause:

Misleading
insights

Poor ROI in
deployment

Damaged
trust in
analytics

A slightly less accurate but more stable model may be more valuable

Lecture
4

Signs You Are Overfitting



Very high accuracy on training data, but low accuracy on test data



Complex decision boundaries that fit every fluctuation in the data



Performance drops when presented with slightly altered inputs



The model is highly sensitive to small changes in the data

Example: A decision tree that perfectly classifies training data with 15+ branches — but performs poorly on new cases

Lecture
4

Why Does Overfitting Happen?



The model is too complex for the amount or quality of data



The dataset is too small, leading to memorization rather than learning



The model trains too long, continuing to reduce training error while harming test performance



There is noise in the data (randomness mistaken for patterns)

Overfitting is more likely when flexibility is high and data is limited.

**Lecture
4****Underfitting, Good Fit, Overfitting (Conceptual)**

- Three scenarios when fitting a model:

Scenario	Description	Performance
Underfitted	Too simple; fails to capture patterns	Poor on both train/test
Good Fit	Captures meaningful patterns; generalizes well	Balanced train/test error
Overfitted	Too complex; memorizes data	Good

Lecture
4

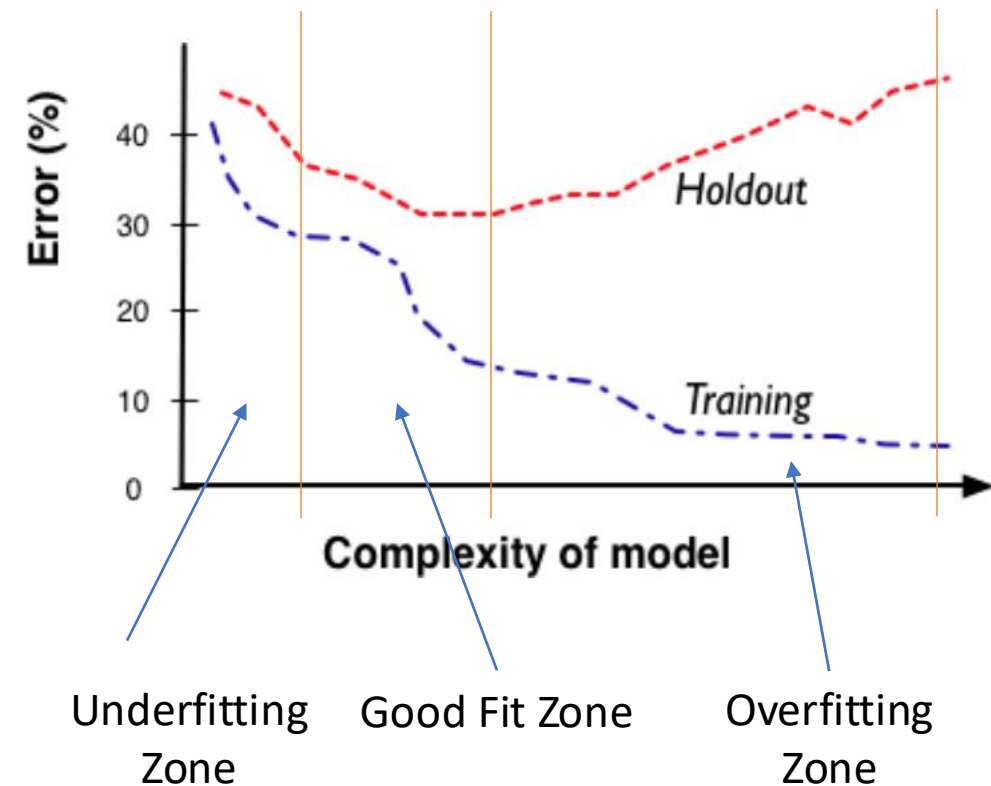
Holdout Data & Fitting Graphs

- **Holdout Data:**
 - A portion of the dataset set aside and not used during training
 - Used only to evaluate how well the model generalizes to new data
 - Prevents the illusion of success from overfitting
- **Fitting Graphs:**
 - Shows model performance vs. complexity, using both Training performance and Holdout (validation) performance

Lecture
4

Fitting Graphs

- What It Reveals:
 - **Underfitting Zone:** Both errors are high $\rightarrow$ model is too simple
 - **Good Fit Zone:** Both errors are low and close
 - **Overfitting Zone:** Training error continues to drop, but holdout error starts increasing
- The optimal model complexity is where the holdout error is minimized — not where the training error is lowest.



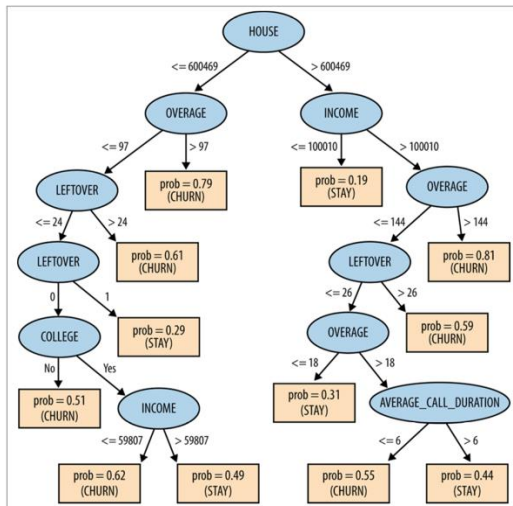
Lecture 4

Overfitting in Decision Trees

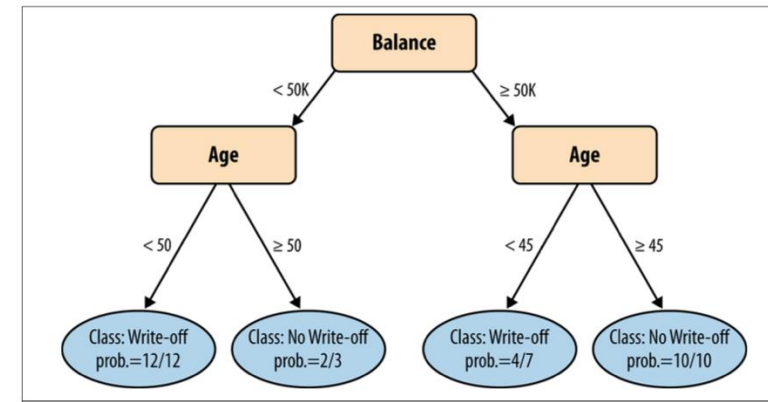
More branches $\neq$ better model

- Deep decision trees can perfectly classify training data
- But the model becomes too specific, losing generality

Deep Tree

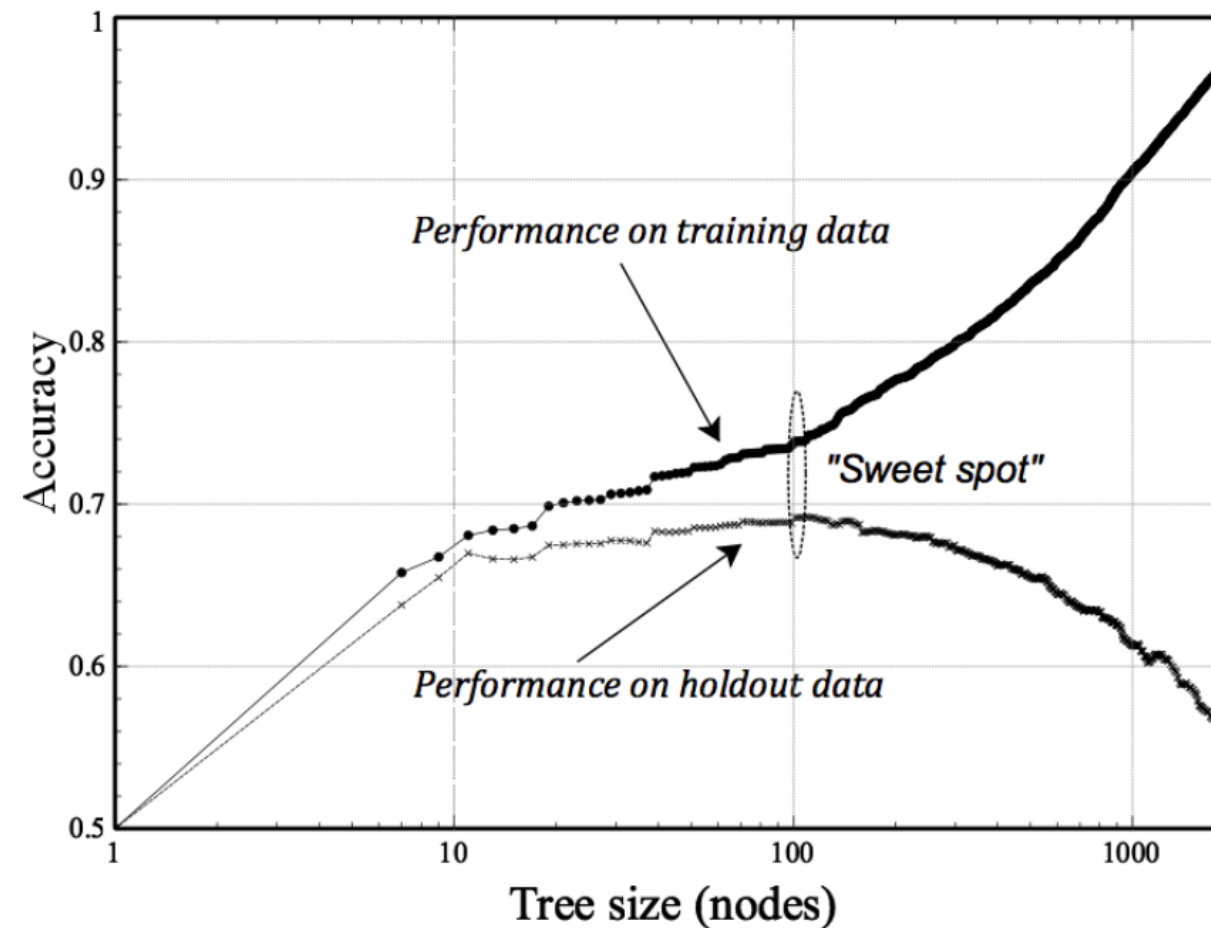


Shallow Tree



Lecture
4

A Typical Fitting Graph for Tree-Induction



Lecture
4

Polynomial Regression

- A polynomial regression model of very high degree fits every data point
- Such models often fails on new inputs
- For example:
 - 2nd-degree curve: smooth and predictive
 - 15th-degree curve: oscillates wildly between points
- High-degree models = high variance = low reliability

Lecture
4

Overfitting Due to Noise

- Models may learn from noise (random, non-repeatable data) → poor generalization → Overfitting mistakes random variation for real structure

- Example:

A model learns that customers from a single ZIP code always churn — when in reality, it was a random coincidence

Lecture
4

High Variance Models

- **Variance**: refers to how much the model changes with different data samples
- Overfit models have high variance → they're unstable
- Small change in input → large change in output
- A good model trades a bit of training error for stability on new data

Models with high variance are unreliable in real-world settings

Lecture
4

Early Warning Signs of Overfitting



Large gap between training and test accuracy



Model performance worsens as complexity increases



Accuracy improves on training but stabilizes or drops on validation



The model seems too confident about trivial patterns

Overfitting is not a training success — it's a deployment failure waiting to happen.

Lecture
4

How to Spot Overfitting Early?

- **Early detection allows you to prune, simplify, or regularize before deployment**

Use visual diagnostics:

- Learning curves (training vs. validation error)
- Complexity vs. performance plots

Monitor performance gap:

- High train accuracy + low test accuracy = red flag

Lecture
4

Training vs. Test Performance

- Overfitting is revealed when training and test accuracy diverge
- A model may perform very well on training data but fail to generalize
- For Example:
 - Training accuracy: 98%
 - Test accuracy: 72%

Always compare model performance across separate datasets

Lecture 4

Performance Gap Warning Signs

Metric	Overfit Model	Well-Generalized Model
Training Accuracy	Very high (95–100%)	High (but not perfect)
Test Accuracy	Much lower (e.g., 60%)	Similar to training
Error Gap	Large	Small and stable

A large gap is MORE important than high accuracy

Lecture 4

Underfitting vs. Overfitting Summary

Type	Cause	Symptom	How to Fix it?
Underfitting	Model too simple	Poor performance on all sets	Add Complexity
Overfitting	Model too complex or noisy	High train, low test performance	Simplify, regularize, or use more data

Lecture
04

Practical Strategies to Avoid or Reduce Overfitting



Lecture
4

How to Avoid Overfitting – Overview

- Overfitting can be avoided using a techniques that:

Reduce model
complexity

Ensure honest
evaluation

Control how
much the model
learns from noise

- Key categories:

Data-driven
evaluation
(holdout sets,
cross-validation)

Complexity control
(regularization,
pruning)

Model training
control (early
stopping, feature
selection)

Lecture
4

How to Avoid Overfitting – Overview

- Common strategies include:

Validation
techniques (e.g.
cross-validation)

Regularization

Pruning and
simplification

Using more data

Avoiding leakage

The goal is not just to fit the data — but to generalize beyond it

Lecture
4

Holdout Data & Cross-Validation

Holdout set:
separate
part of data
not used in
training

Cross-
validation:
rotates
training/test
ing across
folds

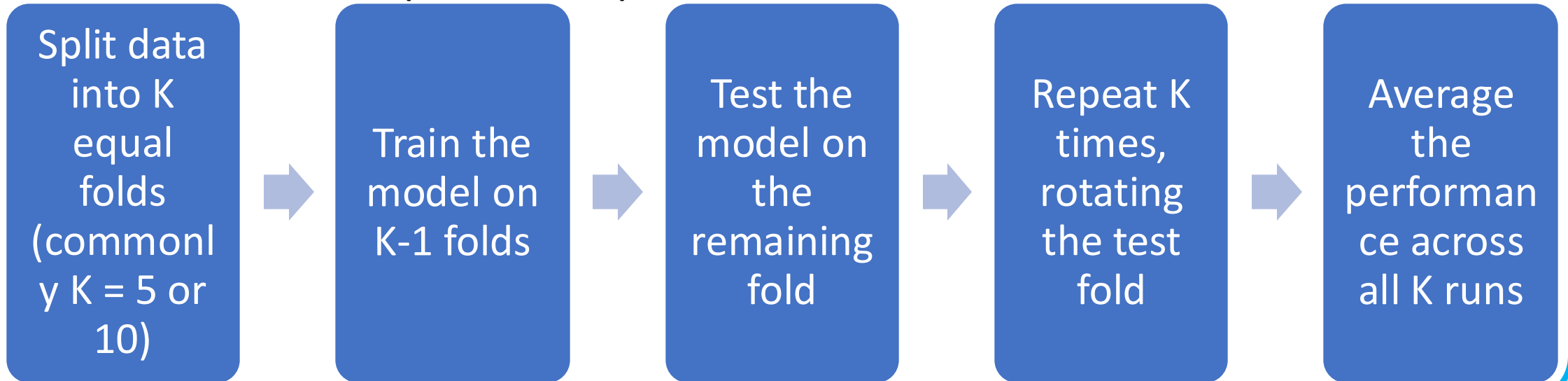
Helps
identify
overfitting
early

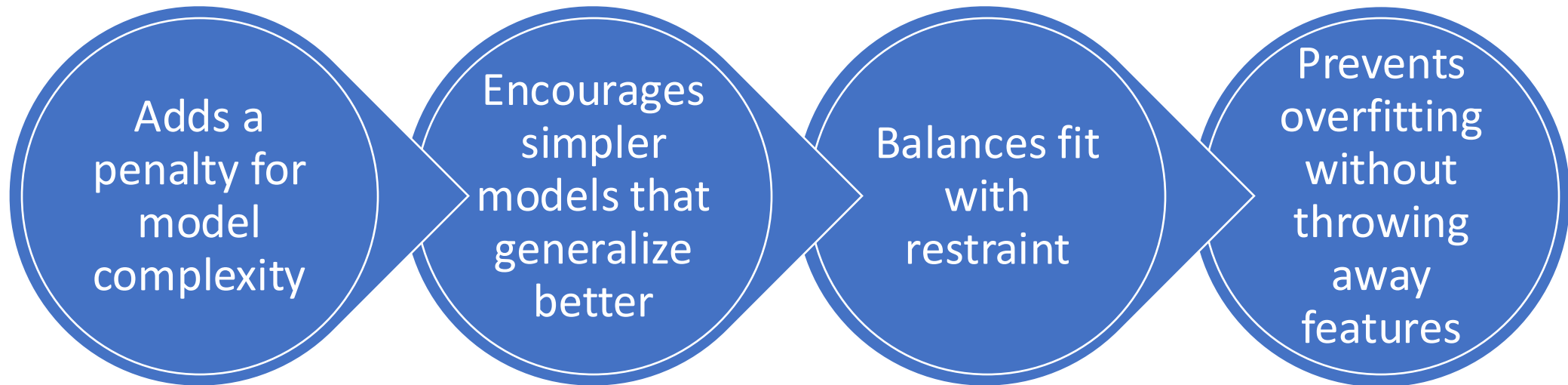
Always test
your model
on unseen
data

Lecture
4

What is Cross-Validation?

- Cross-validation is a technique to assess how well your model generalizes to unseen data by training and testing it on different subsets of the dataset.
- How It Works (K-Fold CV):



**Lecture
4****Regularization Concept**

Lecture
4

L1 (Lasso) vs. L2 (Ridge) Regularization

Type	Description	Best For
L1	Shrinks coefficients to zero	Feature selection
L2	Shrinks all coefficients evenly	Smoothing without zeroing out

- Both techniques reduce variance and help prevent overfitting

**Lecture
4****Early Stopping**

Used in iterative models (e.g., boosting, neural nets)



Stops training when performance on validation set starts to decline



Prevents learning noise or irrelevant patterns



Simple, effective way to control overfitting in deep models

Lecture
4

Pruning (For Decision Trees)

- Trees can grow too deep, capturing noise
- In decision trees, pruning removes branches that do not improve generalization
- Reduces overfitting and Improves interpretability

- Types:

Pre-pruning
(set depth or leaf
limits before
training)

Post-pruning
(grow a full tree,
then trim it based on
validation error)

Lecture
4

Simplification - Limiting Model Complexity

- In many cases, simpler models (linear, shallow trees) perform just as well or better than complex ones.
- Techniques:
 - Use fewer parameters
 - Restrict feature interactions
 - Choose simpler models when possible

**Lecture
4****Use Simpler Models First**

Start with linear/logistic regression before trying deep trees or neural networks

Easier to interpret, often good enough

Simpler models are more robust and faster to train

Complexity should be justified, not assumed

Lecture
4

Ensemble Methods Help

- Combine multiple models to reduce variance thus improving prediction and reducing overfitting.
- Ensembles average out errors and make models more robust.
- Common techniques:
 - **Bagging (e.g., Random Forests)**: Trains models on bootstrapped data subsets and reduces variance.
 - **Boosting (e.g., XGBoost)**: Focuses on previous errors, can reduce bias but must be regularized

**Lecture
4****Feature Selection**

- Remove irrelevant or redundant features that introduce noise, inflate complexity, and increase overfitting risk.
- Techniques:

Filter methods:

Use correlation/statistics to remove irrelevant features

Wrapper methods:

Evaluate subsets of features using model performance

Embedded methods:

Feature selection occurs during training (e.g. Lasso)

Fewer features = less chance to overfit

**Lecture
4**

Dimensionality Reduction Techniques

- When feature count is high, use transformation methods to reduce dimensionality.
- Leads to lower variance, better generalization, and faster training.
- Techniques:

Principal Component Analysis (PCA):
Converts features into orthogonal components

Autoencoders:
Neural networks that learn compressed representations

Lecture
4

Avoiding Data Leakage

Leakage = model trained on future or external info

Example: using “Total Revenue” before it is earned

Can lead to deceptively high training accuracy

Always simulate real-world timing and access

Lecture
4

Avoid Target Leakage

- A special form of leakage where the model indirectly “sees” the answer
- Example: churn_flag_updated column used as a feature to predict churn
- Avoid any variable correlated with the target due to timing or logic

Leakage is the most subtle but severe cause of overfitting

Lecture
4

Overfitting with Small Datasets

- **Overfitting is more likely when:**
 - Sample size is small
 - Feature count is large
- **Consider:**
 - Collecting more data
 - Using regularization
 - Simplifying the model

Data scarcity demands simplicity and discipline

**Lecture
4****Balancing Accuracy and Interpretability**

- Choose based on use case, not just accuracy

Model Type	Accuracy Potential	Interpretability
Decision Tree	Medium	High
Random Forest	High	Medium
Neural Network	Very High	Low
Logistic Regression	Medium	High

Lecture
4

The Role of Domain Knowledge (Expertise)

- Domain knowledge is a defense against blind overfitting
- Helps:
 - Select meaningful features
 - Avoid irrelevant complexity
 - Catch leakage or logic flaws
- Can reduce reliance on complex models



Lecture 4

Summary Table – Strategies to Avoid Overfitting

Strategy	Purpose
Cross-validation	Honest performance evaluation
Regularization (L1/L2)	Penalize complexity
Pruning / Early Stopping	Limit overfitting during training
Feature Selection	Remove irrelevant inputs
Simple Models	Reduce overfitting risk
Avoiding Leakage	Prevent accidental cheating

**Lecture
4**

Monitoring Overfitting in Production

Overfitting can emerge after deployment as:

- Customer behavior shifts
- Data drift
- Feature relevance fades

Best practices:

- Monitor performance over time
- Use A/B testing
- Re-train as needed

Lecture
4

Key Points

Overfitting means your model learns noise, not signal

The best models balance bias and variance

Evaluation must happen on unseen data

Simpler models often generalize better

Overfitting isn't just a technical issue — it affects business decisions

For Discussion Panel and Next Lecture:

- Use the open-source Telco Customer Churn dataset available on [Kaggle](#).
- **Questions:**
 1. what can Overfitting in customer-churn models result in?
 2. Which tools can we use to prevent or reduce overfitting?

Lecture 4



For Discussion Panel and Next Lecture



Thank You