**Lecture Notes: Overfitting & Underfitting**

**1. Why this is important**

* **Common interview question** in data science: “What is overfitting, and how do we deal with it?”
* Knowing this shows you understand the *limitations* of models.

**2. Key Concepts**

* **Underfitting**
  + Model is *too simple*.
  + Cannot capture the underlying pattern in the data.
  + Leads to poor performance both on training and test data.
  + Example: Using a straight line to model a very curvy relationship.
* **Overfitting**
  + Model is *too complex*.
  + Memorizes the training data, including random noise.
  + Performs very well on training data but fails on unseen (test) data.
  + Example: A model that predicts training data almost perfectly but fails in real-world use.

**3. Graph Intuition**

* Imagine a regression problem:
  + **Good fit**: model line is close to the true function, generalizes well.
  + **Underfit (too simple)**: linear model when data clearly has curves → high error.
  + **Overfit (too complex)**: model zigzags through every point, capturing noise → low training error but poor generalization.

**4. Costs of Underfitting**

* High loss function values.
* Low accuracy.
* Model feels “clumsy.”
* Indicates either:
  1. No real relationships exist in the data, or
  2. You need a more complex model.

**5. Costs of Overfitting**

* Training accuracy looks amazing (even >99%).
* But on real-world data (test set), the performance collapses.
* **Example: Forex trading case**
  + Model trained with 50 indicators.
  + Training results: super high accuracy, very low loss.
  + In practice: trades fail → model captured **random investor noise**, not real economic patterns.

**6. Why Overfitting Happens**

* The model “learns” not only the patterns but also the random fluctuations (noise).
* Computers are not wrong:
  + Training looked perfect mathematically.
  + Problem is: the *data itself* had randomness.
* The model didn’t generalize, it memorized.

**7. Practical Challenge**

* In simple 2D graphs, it’s easy to “see” underfitting/overfitting.
* But in real life, datasets often have **dozens or hundreds of dimensions** (features).
* We can’t “visualize” overfitting directly → we must rely on statistical tools & validation methods.

**8. Summary**

* **Underfitting**: model is too simple, misses the pattern.
* **Overfitting**: model is too complex, memorizes noise.
* **Goal**: find the balance → a model that generalizes well.
* Remedies (next lessons): regularization, cross-validation, pruning, dropout, etc.

✅ At this point, you should:

* Be able to define **overfitting** & **underfitting** in an interview.
* Understand their differences with simple regression/classification intuition.
* Recognize that overfitting is dangerous because it looks good in training but fails in real-world tasks.