**AI/ML Interns – Bhavya**

**Task 5 Day6 Learning.**

**Learning Topics:**

* Overfitting V/S Underfitting.
* Cross Validation Basics.
* Model Improvement Techniques.

1. **OverFitting V/S Underfitting:**

Overfitting and underfitting are processes in machine learning that describe how well a model devotes itself in learning from data. They seem to illustrate the two extremes of fitting a model for a given dataset.

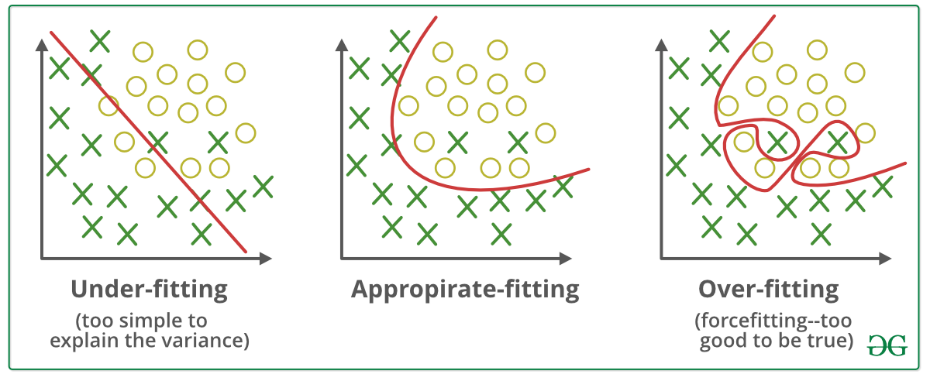
Overfitting occurs when a model learns the training data very well, including the noise extra and specific details therein. It performs well on training data but poorly on test data. A student cramming the answers on a specific test but unable to face a new quiz because the student never really understood the subject areas is a classic analogy used for overfitting.

Example: In your Titanic dataset, the model learns very specific passenger details (e.g., exact age or fares) that are not general patterns and obtains 95% accuracy on the training data but 60% on the test data.

Underfitting happens when the model is too simple to capture any pattern from the training data and performs poorly on both training and test data. Think of a student who does not study at all and is unable to answer the questions correctly.

Example: Consider the logistic regression model based on Pclass only. It probably misses some important patterns (such as Sex or Age), rallying a low accuracy of 50%, both on training and test data.

Striking a balance: Making the model too simple or providing additional data to prevent overfitting. Another solution to underfitting is to select a more complicated model or create better features. A balanced model generalizes well and does equally well on training and test data.



**Techniques to Reduce Underfitting:**

* Increase model complexity.
* Increase the number of features, performing feature engineering.
* Remove noise from the data.
* Increase the number of epochs or increase the duration of training to get better results.

**Techniques to Reduce Overfitting:**

* Improving the quality of training data reduces overfitting by focusing on meaningful patterns, mitigate the risk of fitting the noise or irrelevant features.
* Increase the training data can improve the model's ability to generalize to unseen data and reduce the likelihood of overfitting.
* Reduce model complexity.
* Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
* Ridge Regularization and Lasso Regularization.
* Use dropout for neural networks to tackle overfitting.

| **Aspect** | **Underfitting** | **Overfitting** |
| --- | --- | --- |
| **Learning Style** | Under-learns (too simple) | Over-memorizes (too complex) |
| **Training Performance** | Poor (high error) | Excellent (very low error) |
| **New Data Performance** | Poor (high error) | Poor (high error) |
| **Model Complexity** | Too simple (few parameters) | Too complex (many parameters) |
| **Solution** | Add more features/complexity | Reduce complexity/add more data |

1. **Cross Validation Basics:**

Definition: Cross-validation tests how well a machine learning model performs on new, unseen data by splitting the data into subsets, training the model on some, testing on others multiple times, and then averaging the results to have a solid estimate of the model's performance.

Types:

**Holdout:** Data is split into 50-50 (train/test). It is simple but can miss some important patterns (high bias).

**LOOCV:** Train on all data points except one, test on that one, and repeat for all points. Low bias but slow and sensitive to outliers.

**Stratified K-Fold**: Similar to K-Fold but maintains class ratios (e.g., % survivors) in each fold. Good for imbalanced data.

**K-Fold:** Divide the data into k parts, train on k-1 parts, test on 1 part, and repeat this process k times (k=10 is a popular choice).

Cross-validation protects against overfitting so that the model indeed generalizes well.

Real-World Example

Imagine you are trying to distinguish between cats and dogs:

Without Cross Validation: Take 100 photos and train and 20 photos for testing (just 1 testing process)

With Cross Validation: Split the 120 photos into 5 groups, train on 4 groups, test on 1 group, and do this 5 times for 5 different groups as test groups.

1. **Model Improvement Techniques:**

Model improvement can be likened to refining a recipe or adjusting an instrument—continuous tweaks lead to better results. In machine learning, improving a model involves systematically enhancing its predictive performance through several established techniques. The following sections outline these methods.

**1. Data Quality**

* **Cleaning the Data**
  + Remove errors, duplicates, and missing values.
  + Example: Exclude house listings with unrealistic prices or improbable features (e.g., $0 price or 800 bedrooms).
* **Increasing Data Quantity**
  + Expand the dataset to provide the model with more examples, which can facilitate more robust learning.

**2. Feature Engineering**

* **Creating New Features**
  + Derive new variables from existing data to capture additional information.
  + Example: Generate “age” from “birth year,” or calculate BMI from height and weight.
* **Removing Unhelpful Features**
  + Eliminate features that do not contribute meaningfully to model performance.

**3. Algorithm Selection**

* **Experimenting with Different Algorithms**
  + Test various machine learning algorithms to identify the most suitable for the task.
  + Example: If Linear Regression achieves 70% accuracy, consider Random Forest or Neural Networks to potentially improve outcomes.

**4. Hyperparameter Tuning**

* **Adjusting Model Settings**
  + Fine-tune parameters such as learning rate or tree depth to optimize model performance.
  + Example: Modify the learning rate from 0.01 to 0.001, or increase tree depth from 5 to 10.