<u>WEEK 2: MACHINE LEARNING CONCEPTS</u>

DAY 10 (04/07/2025)

Feature Engineering and Hyperparameter Tuning:

Once the model and evaluation setup are ready, the next goal is to **improve performance** using better features and parameter optimization.

1. Feature Engineering

Features are the inputs the model uses to make predictions. Creating effective features can often boost performance more than changing the algorithm itself.

Key Techniques:

- **Feature Selection:** Choose only relevant features to reduce noise.

- **Feature Creation:** Combine or transform existing data into new, meaningful variables.

- **Normalization & Scaling:** Ensures all features contribute equally.

- **Dimensionality Reduction:** Compress large feature sets while keeping key information.

Example:

Features like "Temperature Difference per Day" or "Average Rainfall in Last 7 Days" can help predict rainfall more accurately than just using raw daily values.

2. Hyperparameters vs Parameters

- **Parameters:** Learned automatically during training (like weights in a neural network).

- **Hyperparameters:** Set manually before training (like learning rate, or batch size).

3. Hyperparameter Tuning

Hyperparameter tuning is about adjusting the settings of a model before training so that it works as well as possible.

These settings (called **hyperparameters**) are not learned by the model — we choose them manually. Examples include the learning rate in a neural network, or the depth of a tree in a decision tree.

By carefully choosing these values, we can **improve the model's accuracy** and make it perform better on new data.

4. Regularization

Regularization is a technique used to **prevent a model from overfitting** — that is, learning the training data too perfectly and failing on new data.

- **Why it's needed:**
  When a model is too complex, it may memorize all the training examples, including noise or outliers, instead of learning general patterns. Regularization discourages the model from becoming too complex.

- **How it works:**
  Regularization adds a **penalty** for large weights or complex relationships in the model. This forces the model to focus on the **most important features** and ignore irrelevant ones.

- **Effect:**
  The model becomes simpler, more general, and performs better on unseen data.

Example:
Imagine predicting house prices: without regularization, the model might overemphasize unusual properties (like a rare color or a weird shaped garden). With regularization, the model focuses on key features like size, location, and number of rooms, giving more reliable predictions.

Reflection

Today emphasized that fine-tuning and thoughtful feature design can turn an average model into an exceptional one.
Hyperparameter tuning is like adjusting knobs — small changes can make a huge difference in the final outcome.