Basic terminology:

**1. Features and Labels**

* **Features**: These are the input variables used to predict the outcome. Features are the "data" that goes into the model.
* **Labels**: The output or the target variable that we want to predict. It's what the model tries to learn from the features.

**Example**:

* Imagine you're building a model to predict **house prices** based on various factors.
  + **Features**: Size of the house (in square feet), number of bedrooms, neighborhood, age of the house.
  + **Label**: Price of the house (the value you're trying to predict).

Here, the features are all the data points (size, number of bedrooms, etc.) that you use to predict the price (label).

**2. Training and Testing**

* **Training**: This is the process where the model learns from the training data (features and labels). The model adjusts its parameters to minimize errors in predictions.
* **Testing**: After training, you evaluate the model's performance using the test data, which it hasn’t seen before. This helps you understand how well it generalizes to new, unseen data.

**Example**:

* Suppose you have a dataset of 1000 houses, and you split it into 800 houses for **training** and 200 houses for **testing**.
  + **Training Data**: The model learns from 800 houses.
  + **Testing Data**: After training, you check how well the model predicts house prices for the remaining 200 houses that it hasn't seen before.

**3. Overfitting**

* Overfitting happens when the model learns the details and noise of the training data to an extent that it negatively impacts its performance on new data (the test data). Essentially, the model becomes "too good" at predicting the training data but fails to generalize to unseen data.

**Example**:

* Imagine you have a model predicting house prices, but the model learns very specific details about the 800 houses in the training data, like how a certain rare event (e.g., a celebrity buying a house) affected the price. If the model is overly complex, it might fit these specifics very well but fail to predict accurately for the 200 test houses because those events don't happen for new houses.

**Signs of Overfitting**: The model does great on training data but poorly on test data.

**4. Underfitting**

* Underfitting occurs when the model is too simple to capture the underlying patterns in the data. This results in poor performance on both the training data and the test data.

**Example**:

* If you're trying to predict house prices but use only one feature (like just the size of the house) and ignore other important features (like location, number of rooms), your model might be too simple to make accurate predictions.

**Signs of Underfitting**: The model performs poorly on both the training and test data because it can't capture the complexity of the problem.

**Putting it all together:**

* Imagine you are using a **linear regression model** to predict house prices.
  + **Features**: Size, number of bedrooms, etc.
  + **Label**: Price
  + **Training**: The model learns from historical house data to understand the relationship between features and prices.
  + **Testing**: The model's predictions are evaluated on unseen house data.

If the model is too complex (too many features or parameters), it might overfit and learn irrelevant details from the training data. If it's too simple, it might underfit and fail to capture important patterns.

The goal is to find the right balance, where the model learns enough to generalize well but doesn't memorize the training data.