Hyperparameter Tuning: Search

Hyperparameter tuning is the process of finding the optimal combination of hyperparameters for a machine learning model. A hyperparameter search consists of four key components:

1. **Hyperparameter space**: This defines the range of possible values for each hyperparameter you want to tune. For example, for a random forest, this might include the number of trees, maximum depth, minimum samples per leaf, etc.
2. **Sampling method**: This determines how candidate hyperparameter combinations are selected from the space. Common approaches include:
    - Grid search (exhaustively tries all combinations)
    - Random search (randomly samples combinations)
    - Bayesian optimization (intelligently selects promising combinations based on previous results)
3. **Cross-validation scheme**: This is how you evaluate each candidate hyperparameter set. Typically k-fold cross-validation is used, where the data is split into k parts, with each part serving as the test set once while the others are used for training.
4. **Performance metric**: This is what you're trying to optimize (either minimize or maximize). The choice depends on your problem - accuracy, precision, recall, RMSE, etc.

Metrics

Metrics are quantitative measures used to evaluate model performance. They vary by problem type:

Classification Metrics (for categorical outcomes):

- **Accuracy**: Percentage of correct predictions
- **Precision**: Percentage of positive predictions that were correct
- **Recall**: Percentage of actual positives that were correctly identified
- **F1-score**: Harmonic mean of precision and recall

- **ROC-AUC**: Area under the receiver operating characteristic curve

Regression Metrics (for continuous outcomes):

- **Mean Squared Error (MSE)**: Average of squared errors
- **Root Mean Squared Error (RMSE)**: Square root of MSE
- **Mean Absolute Error (MAE)**: Average of absolute errors
- **R-squared**: Proportion of variance explained by the model

In Scikit-learn:

Scikit-learn provides these metrics in its `metrics` module. They're used by:

1. Importing the relevant metric function
2. Passing true values and predictions to the function
3. For hyperparameter tuning, the metric is specified in the search object (like GridSearchCV)

Creating Custom Metrics:

You can define your own metrics when standard ones don't capture your specific needs. This involves:

1. Creating a function that takes true values and predictions
2. Implementing your custom calculation
3. Ensuring it returns a single numerical value (higher=better or lower=better)
4. Using it in your model evaluation or hyperparameter search