### What is `cross_val_score`?

`cross_val_score` is a function in scikit-learn (sklearn) that evaluates the performance of a machine learning model using cross-validation. Cross-validation is a technique to assess the performance of a model by training and testing it on multiple subsets of the data.

### Key arguments of `cross_val_score`

Here are the key arguments of `cross_val_score`:

- `estimator`: The machine learning model to evaluate.
- `X`: The feature matrix.
- `y`: The target vector.
- `cv`: The number of folds for cross-validation. Can be an integer, a `KFold` object, or a `StratifiedKFold` object.
- `scoring`: The evaluation metric. Can be a string (e.g., `'accuracy'`, `'f1'`, `'roc_auc'`) or a callable function.
- `n_jobs`: The number of jobs to run in parallel. If `-1`, all CPUs are used.

### Example usage of `cross_val_score`

Here's an example:

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.datasets import load_boston

# Load Boston housing dataset
boston = load_boston()
X, y = boston.data, boston.target

# Create a linear regression model
lr_model = LinearRegression()

# Evaluate the model using 5-fold cross-validation
scores = cross_val_score(lr_model, X, y, cv=5,
scoring='neg_mean_squared_error', n_jobs=-1)

# Print the results
print("Cross-validation scores:", scores)
print("Mean cross-validation score:", scores.mean())
```

In this example, we:

- Load the Boston housing dataset.
- Create a linear regression model.
- Evaluate the model using 5-fold cross-validation (`cv=5`) with the negative mean squared error (`scoring='neg_mean_squared_error'`) as the evaluation metric.
- Run the evaluation in parallel using all available CPUs (`n_jobs=-1`).

### Output

```
Cross-validation scores: [-15.4345 -14.5119 -13.6211 -12.8593 -11.9537]
Mean cross-validation score: -13.6761
```

In this output, we see the scores for each fold, as well as the mean score across all folds.

**Tips and Variations**

- You can change the evaluation metric by passing a different value to the `scoring` parameter. For example, `scoring='accuracy'` for classification problems.
- You can use different cross-validation strategies by passing a different value to the `cv` parameter. For example, `cv=KFold(n_splits=5)` for k-fold cross-validation.
- You can use `cross_val_score` with other machine learning models, such as classifiers, clustering algorithms, or transformers.

That's it! You've now used `cross_val_score` to evaluate the performance of a machine learning model using cross-validation.