Training, Validation, and Test Sets:

1. Training Set: The dataset used to train the model.
2. Validation Set: The dataset used to validate the model's performance during training and fine-tune hyperparameters.
3. Test Set: The dataset used for the final evaluation of the model to measure its generalization performance.

Best Practices

- Always split your dataset into training, validation, and test sets to avoid overfitting and biased results.
- Shuffle your data before splitting to ensure a balanced distribution of classes across the sets.
- Use a suitable validation method based on the size and nature of your dataset (e.g., hold-out validation, K-fold cross-validation, or iterated K-fold validation with shuffling).

Hold-out Validation

1. Shuffle your data.
2. Split the data into training, validation, and test sets.
3. Train the model on the training set and evaluate it on the validation set.
4. Fine-tune the model based on the validation set performance.
5. Train the final model on the combined training and validation sets.
6. Evaluate the model on the test set.

Code :

```
import numpy as np

num_validation_samples = 10000
np.random.shuffle(data)
validation_data = data[:num_validation_samples]
data = data[num_validation_samples:]
training_data = data[:]

model = get_model()
model.train(training_data)
validation_score = model.evaluate(validation_data)
```

```
model = get_model()
model.train(np.concatenate([training_data, validation_data]))
test_score = model.evaluate(test_data)
```

K-fold Cross-validation

1. Divide the dataset into K equally sized folds.
2. Train the model K times, each time using K-1 folds for training and the remaining fold for validation.
3. Calculate the average performance across all K iterations.

Code:

```from sklearn.model_selection import cross_val_score

from sklearn.linear_model import LogisticRegression

from sklearn.datasets import load_iris

iris = load_iris()

X, y = iris.data, iris.target

model = LogisticRegression()

scores = cross_val_score(model, X, y, cv=5)

average_accuracy = scores.mean()

```

Iterated K-fold Validation with Shuffling

1. Repeat K-fold cross-validation for a specified number of iterations.

2. Shuffle the data before each iteration.
3. Calculate the average performance across all iterations.

Remember to choose the validation method that best suits your dataset size and nature, and make sure to evaluate your model on a test set that it has never seen before to measure its generalization performance accurately.