```python
import numpy as np

def knn_regression(X_train, y_train, X_test, k=3):
    """
    Implements a KNN algorithm for regression tasks, predicting the average of the k nearest neighbors.

    Args:
        X_train (numpy.ndarray): Training feature data.
        y_train (numpy.ndarray): Training target values.
        X_test (numpy.ndarray): Test feature data.
        k (int): Number of nearest neighbors to consider.

    Returns:
        numpy.ndarray: Predicted target values for the test data.
    """

    y_pred = []
    for x_test in X_test:
        distances = np.linalg.norm(X_train - x_test, axis=1)  # Calculate Euclidean distances
        nearest_indices = np.argsort(distances)[:k]  # Get indices of k nearest neighbors
        predicted_value = np.mean(y_train[nearest_indices])  # Calculate average of target values
        y_pred.append(predicted_value)

    return np.array(y_pred)

# Example usage with a small dataset
X_train = np.array([[1, 2], [3, 4], [5, 6], [7, 8]])
y_train = np.array([1.5, 3.5, 5.5, 7.5])
X_test = np.array([[2, 3]])

predictions = knn_regression(X_train, y_train, X_test, k=2)
print("Predicted value:", predictions)
```