

```
import AVFAudio
import AVFoundation
import Accelerate
import Combine
import CoreGraphics
import CoreImage
import CoreML
import CoreMedia
import CoreVideo
import Foundation
import OSLog
import RegexBuilder
import SoundAnalysis
import TabularData
import UniformTypeIdentifiers
import Vision
import _Concurrency
import _StringProcessing
import _SwiftConcurrencyShims
import os

/// A batch of annotated examples for fitting a supervised estimator.
@available macos 15.0  ios 18.0  tvos 18.0  visionos 2.0
watchos 11.0
public struct AnnotatedBatch<Scalar> where Scalar : MLShapedArrayScalar

    /// The shaped array of features.
    public var features : MLShapedArray<Scalar>

    /// The shaped array of annotations.
    public var annotations : MLShapedArray<Scalar>

    /// The number of examples in the batch.
    ///
    /// This is the first dimension of both features and annotations.
    public var count : Int {
        get
        /// Creates an annotated batch.
        ///
        /// The features and annotations must have the same rank, and the first
        /// dimensions must be equal.
        ///
        /// - Parameters:
        ///   - features: A shaped array of features.
        ///   - annotations: A shaped array of annotations.
        public init(features: MLShapedArray<Scalar>, annotations: MLShapedArray<Scalar>)
    }
}
```

```
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
extension AnnotatedBatch Encodable where Scalar Encodable
```

```
    /// Encodes this value into the given encoder.  
    ///  
    /// If the value fails to encode anything, `encoder` will encode an empty  
    /// keyed container in its place.  
    ///  
    /// This function throws an error if any values are invalid for the given  
    /// encoder's format.  
    ///  
    /// - Parameter encoder: The encoder to write data to.  
public func encode any Encoder throws
```

```
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
extension AnnotatedBatch Decodable where Scalar Decodable
```

```
    /// Creates a new instance by decoding from the given decoder.  
    ///  
    /// This initializer throws an error if reading from the decoder fails, or  
    /// if the data read is corrupted or otherwise invalid.  
    ///  
    /// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws
```

```
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
extension AnnotatedBatch Equatable where Scalar Equatable
```

```
    /// Returns a Boolean value indicating whether two values are equal.  
    ///  
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    ///  
    /// - Parameters:  
    /// - lhs: A value to compare.  
    /// - rhs: Another value to compare.  
public static func AnnotatedBatch Scalar  
AnnotatedBatch Scalar Bool
```

```
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
extension AnnotatedBatch Sendable where Scalar Sendable
```

```
/// An annotated example for fitting a supervised estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct AnnotatedFeature Feature Annotation
```

```
/// The feature value.
```

```
public var feature Feature
```

```
/// The annotation.
```

```
public var annotation Annotation
```

```
/// Creates an example with a feature and an annotation.
```

```
public init Feature Annotation
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension AnnotatedFeature Encodable where Feature  
Encodable Annotation Encodable
```

```
/// Encodes this value into the given encoder.
```

```
///
```

```
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.
```

```
///
```

```
/// This function throws an error if any values are invalid for the given  
/// encoder's format.
```

```
///
```

```
/// - Parameter encoder: The encoder to write data to.
```

```
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension AnnotatedFeature Decodable where Feature  
Decodable Annotation Decodable
```

```
/// Creates a new instance by decoding from the given decoder.
```

```
///
```

```
/// This initializer throws an error if reading from the decoder fails, or  
/// if the data read is corrupted or otherwise invalid.
```

```
///
```

```
/// - Parameter decoder: The decoder to read data from.
```

```
public init any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
extension AnnotatedFeature Equatable where Feature  
Equatable Annotation Equatable

    /// Returns a Boolean value indicating whether two values are equal.  

    ///  

    /// Equality is the inverse of inequality. For any values `a` and `b`,  

    /// `a == b` implies that `a != b` is `false`.  

    ///  

    /// - Parameters:  

    /// - lhs: A value to compare.  

    /// - rhs: Another value to compare.  

public static func AnnotatedFeature Feature  
Annotation AnnotatedFeature Feature Annotation Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFeature Hashable where Feature  
Hashable Annotation Hashable

    /// Hashes the essential components of this value by feeding them into the  

    /// given hasher.  

    ///  

    /// Implement this method to conform to the `Hashable` protocol. The  

    /// components used for hashing must be the same as the components  

    compared  

    /// in your type's `==` operator implementation. Call  

`hasher.combine(_:)`  

    /// with each of these components.  

    ///  

    /// - Important: In your implementation of `hash(into:)`,  

    /// don't call `finalize()` on the `hasher` instance provided,  

    /// or replace it with a different instance.  

    /// Doing so may become a compile-time error in the future.  

    ///  

    /// - Parameter hasher: The hasher to use when combining the  

components  

    /// of this instance.  

public func hash inout Hasher

    /// The hash value.  

    ///  

    /// Hash values are not guaranteed to be equal across different executions of  

    /// your program. Do not save hash values to use during a future execution.  

    ///  

    /// - Important: `hashValue` is deprecated as a `Hashable`  

requirement. To  

    /// conform to `Hashable`, implement the `hash(into:)` requirement  

instead.  

    /// The compiler provides an implementation for `hashValue` for you.  

public var hashValue Int get
```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFeature Sendable where Feature
Sendable Annotation Sendable

/// An adaptor that converts a regular estimator to a tabular estimator by selecting
features and annotations from columns.
///
/// Tabular estimators use multiple features columns as input. When there is a
single column of features, you may use
/// a non-tabular estimator. Do this by combining multiple columns with a
`ColumnConcatenator` transformer. Once
/// there is a single column of features, use `AnnotatedFeatureProvider` to
specify which column contains the features,
/// which column contains the annotations, and which column should hold the
results.
///
/// When using `AnnotatedFeatureProvider`, make sure to handle missing
values before using a non-tabular estimator
/// that takes non-optional values. This example includes an
`OptionalUnwrapper` transformer.
///
///     let concatenation = ColumnConcatenator<Float>(
///         columnSelection: .include(columnNames: ["type",
/// "region"]),
///         concatenatedColumnName: "features"
///     )
///     let regression = AnnotatedFeatureProvider(
///
OptionalUnwrapper<MLShapedArray<Float>>().appending(LinearRegr
essor<Float>()),
///     annotationsColumnName: "price",
///     featuresColumnName: "features",
///     resultsColumnName: "result"
/// )
///     let task = concatenation.appending(regression)
///

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct AnnotatedFeatureProvider Base UnwrappedInput
SupervisedTabularEstimator where Base: SupervisedEstimator
Base Transformer Input UnwrappedInput

/// The transformer type created by this estimator.
public typealias Transformer
ColumnSelectorTransformer Base Transformer UnwrappedInput

/// The annotation type.
public typealias Annotation Base Annotation

```

```

    ///> The base estimator.
public var base Base

    ///> The annotation column identifier.
public var annotationColumnID ColumnID AnnotatedFeatureProvider Base
UnwrappedInput Annotation

    ///> The features column name.
public var featuresColumnName String

    ///> The results column name.
public var resultsColumnName String

    ///> Creates an adaptor that converts a regular estimator to a tabular
estimator.
    ///
    ///> - Parameters:
    ///>   - base: A supervised estimator.
    ///>   - annotationsColumnName: The annotations column name.
    ///>   - featuresColumnName: The features column name.
    ///>   - resultsColumnName: The results column name.
public init _ Base String
"targets" String "features"
String "results"

    ///> Fits a transformer to a data frame
    ///
    ///> - Parameters:
    ///>   - input: A data frame containing examples used for fitting the
transformer.
    ///>   - validation: A data frame containing examples used for
validating the fitted transformer.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
public func fitted DataFrame
    DataFrame nil EventHandler
nil async throws
ColumnSelectorTransformer Base Transformer UnwrappedInput

    ///> Encodes a fitted transformer.
public func encode_
AnnotatedFeatureProvider Base UnwrappedInput Transformer
inout any EstimatorEncoder throws

    ///> Decodes a previously fitted transformer.
public func decode inout any
EstimatorDecoder throws AnnotatedFeatureProvider Base
UnwrappedInput Transformer

```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension AnnotatedFeatureProvider
UpdatableSupervisedTabularEstimator where Base
UpdatableSupervisedEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func makeTransformer
AnnotatedFeatureProvider Base UnwrappedInput Transformer

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///     - transformer: A transformer to update.
    ///     - input: A sequence of examples.
    ///     - eventHandler: An event handler.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func update _ inout
AnnotatedFeatureProvider Base UnwrappedInput Transformer
                    DataFrame           EventHandler      nil
async throws

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///     - transformer: A transformer this estimator creates.
    ///     - encoder: An encoder.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func encodeWithOptimizer _ inout any
AnnotatedFeatureProvider Base UnwrappedInput Transformer
                    EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func decodeWithOptimizer _ inout any
EstimatorDecoder throws AnnotatedFeatureProvider Base
UnwrappedInput Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
extension AnnotatedFeatureProvider Sendable where Base  
Sendable

/// An annotated files collection.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct AnnotatedFiles Collection

/// A type representing the sequence's elements.
public typealias Element AnnotatedFeature URL String

/// A type that represents a position in the collection.
///
/// Valid indices consist of the position of every element and a
/// "past the end" position that's not valid for use as a subscript
/// argument.
public typealias Index
Array AnnotatedFiles Element Index

/// The position of the first element in a nonempty collection.
///
/// If the collection is empty, `startIndex` is equal to `endIndex`.
public var startIndex AnnotatedFiles Index get

/// The collection's "past the end" position---that is, the position one
/// greater than the last valid subscript argument.
///
/// When you need a range that includes the last element of a collection, use
/// the half-open range operator (`..) with `endIndex`. The `..
operator
/// creates a range that doesn't include the upper bound, so it's always
/// safe to use with `endIndex`. For example:
///
///     let numbers = [10, 20, 30, 40, 50]
///     if let index = numbers.firstIndex(of: 30) {
///         print(numbers[index ..< numbers.endIndex])
///     }
///     // Prints "[30, 40, 50]"
///
/// If the collection is empty, `endIndex` is equal to `startIndex`.
public var endIndex AnnotatedFiles Index get

/// Accesses the element at the specified position.
///
/// The following example accesses an element of an array through its
/// subscript to print its value:
///
///     var streets = ["Adams", "Bryant", "Channing",
"Douglas", "Evarts"]
```

```

    /**
     *   print(streets[1])
     *   // Prints "Bryant"
     */
    /**
     * You can subscript a collection with any valid index other than the
     * collection's end index. The end index refers to the position one past
     * the last element of a collection, so it doesn't correspond with an
     * element.
     */
    /**
     * - Parameter position: The position of the element to access.
`position`
    /**
     * must be a valid index of the collection that is not equal to the
     * `endIndex` property.
    /**
    /**
     * - Complexity: O(1)
public subscript           AnnotatedFiles Index
IndexingIterator AnnotatedFiles Element get

    /**
     * Returns the position immediately after the given index.
    /**
    /**
     * The successor of an index must be well defined. For an index `i` into a
     * collection `c`, calling `c.index(after: i)` returns the same index
every
    /**
     * time.
    /**
    /**
     * - Parameter i: A valid index of the collection. `i` must be less than
     * `endIndex`.
    /**
     * - Returns: The index value immediately after `i`.
public func index           AnnotatedFiles Index
AnnotatedFiles Index

    /**
     * A type that represents the indices that are valid for subscripting the
     * collection, in ascending order.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Indices DefaultIndices AnnotatedFiles

    /**
     * A type that provides the collection's iteration interface and
     * encapsulates its iteration state.
    /**
    /**
     * By default, a collection conforms to the `Sequence` protocol by
     * supplying `IndexingIterator` as its associated `Iterator`
     * type.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Iterator
IndexingIterator AnnotatedFiles

    /**
     * A collection representing a contiguous subrange of this collection's
     * elements. The subsequence shares indices with the original collection.

```

```

////
/// The default subsequence type for collections that don't define their own
/// is `Slice`.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias SubSequence Slice AnnotatedFiles

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles

    /// Reads training examples from a directory containing files having their
    labels in the name.
    /// The name can contain multiple words separated by a `separator`. So
    the `index` tells the
        /// position of the label in the file name.
        /// Files with incorrect name format are ignored.
    ///
    /// Take for example this directory structure:
    ///
    ///
    /      /
    /      fold1-foo-file1.png
    /      fold1-foo-file2.png
    /      fold2-foo-file3.png
    /      fold1-bar-file4.png
    /      fold1-bar-file5.png
    /      fold2-bar-file6.png
    //

    /// When we specify separator as "-" and index as 1, it would produce two
    labels (foo and bar) with three
    /// URLs each.
    /// - Parameters:
    ///   - url: URL of directory containing the files.
    ///   - separator: The separator used in the name. Default value is ":".
    ///   - index: Index of the label in the file name. Default value is 0.
    ///   - type: Type of files.
    ///   - continueOnFailure: A Boolean value indicating whether to
    continue reading files after
    /// encountering a file that is not readable. The default value is `false`.
    public init URL
Character ":" Int 0 UTTType
                    Bool false throws

    /// Reads training examples from a directory containing files in labeled sub-
    directories.
    ///
    /// Take for example this directory structure:
    ///
    /      /
    /      foo/

```

```
///           file1.png
///           file2.png
///           bar/
///           file3.png
///           file4.png
///
/// It would produce two labels (foo and bar) with two URLs each.
/// - Parameters:
///   - url: URL of directory containing the files.
///   - type: Type of files.
///   - continueOnFailure: A Boolean value indicating whether to
continue reading files after
///     encountering a file that is not readable. The default value is `false`.
public init                                URL
UTType                           Bool  false  throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles Encodable

/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode          any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles Decodable

/// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.
///
/// - Parameter decoder: The decoder to read data from.
public init          any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles Equatable

/// Returns a Boolean value indicating whether two values are equal.
///
```

```

    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func AnnotatedFiles
AnnotatedFiles Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles Hashable

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
    /// in your type's `==` operator implementation. Call
    `hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,
    /// don't call `finalize()` on the `hasher` instance provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
    components
    /// of this instance.
public func hash inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
    requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
    instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension AnnotatedFiles Sendable

```

```
/// An annotated prediction.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
public struct AnnotatedPrediction Prediction Annotation  
  
/// The predicted value.  
public var prediction Prediction  
  
/// The ground truth annotation.  
public var annotation Annotation  
  
/// Creates an annotated prediction.  
public init Prediction  
Annotation  
  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
extension AnnotatedPrediction Encodable where Prediction  
Encodable Annotation Encodable  
  
/// Encodes this value into the given encoder.  
///  
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.  
///  
/// This function throws an error if any values are invalid for the given  
/// encoder's format.  
///  
/// - Parameter encoder: The encoder to write data to.  
public func encode any Encoder throws  
  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
extension AnnotatedPrediction Decodable where Prediction  
Decodable Annotation Decodable  
  
/// Creates a new instance by decoding from the given decoder.  
///  
/// This initializer throws an error if reading from the decoder fails, or  
/// if the data read is corrupted or otherwise invalid.  
///  
/// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws  
  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
extension AnnotatedPrediction Equatable where Prediction  
Equatable Annotation Equatable  
  
/// Returns a Boolean value indicating whether two values are equal.
```

```


    /**
     * Equality is the inverse of inequality. For any values `a` and `b`,
     * `a == b` implies that `a != b` is `false`.
     */
    /**
     * - Parameters:
     *   - lhs: A value to compare.
     *   - rhs: Another value to compare.
     */
    public static func AnnotatedPrediction Prediction
Annotation AnnotatedPrediction Prediction Annotation
Bool

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension AnnotatedPrediction Hashable where Prediction
Hashable Annotation Hashable

    /**
     * Hashes the essential components of this value by feeding them into the
     * given hasher.
     */
    /**
     * Implement this method to conform to the `Hashable` protocol. The
     * components used for hashing must be the same as the components
     * compared
     * in your type's `==` operator implementation. Call
     `hasher.combine(_:)`
     * with each of these components.
     */
    /**
     * - Important: In your implementation of `hash(into:)`,
     * don't call `finalize()` on the `hasher` instance provided,
     * or replace it with a different instance.
     * Doing so may become a compile-time error in the future.
     */
    /**
     * - Parameter hasher: The hasher to use when combining the
     * components
     * of this instance.
     */
    public func hash inout Hasher

    /**
     * The hash value.
     */
    /**
     * Hash values are not guaranteed to be equal across different executions of
     * your program. Do not save hash values to use during a future execution.
     */
    /**
     * - Important: `hashValue` is deprecated as a `Hashable`
     * requirement. To
     * conform to `Hashable`, implement the `hash(into:)` requirement
     * instead.
     */
    /**
     * The compiler provides an implementation for `hashValue` for you.
     */
    public var hashValue Int get


```

@available **macOS** 14.0 **iOS** 17.0 **tvOS** 17.0 **watchOS** 11.0

```
extension AnnotatedPrediction Sendable where Prediction  
Sendable Annotation Sendable

/// A type-erased async iterator.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct AnyTemporalIterator Element  
AsyncIteratorProtocol

    /// Asynchronously advances to the next element and returns it, or ends the  
    /// sequence if there is no next element.  
    ///  
    /// - Returns: The next element, if it exists, or `nil` to signal the end of  
    /// the sequence.  
public func next async throws Element

/// A type-erased temporal sequence.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct AnyTemporalSequence Feature TemporalSequence

    /// The type of element produced by this asynchronous sequence.  
public typealias Element TemporalFeature Feature

    /// The type of asynchronous iterator that produces elements of this  
    /// asynchronous sequence.  
public typealias AsyncIterator  
AnyTemporalIterator AnyTemporalSequence Feature Element

    /// The number of elements in the sequence if available, calculated  
nondestructively.  
public let count Int

    public init S _ S where Feature S Feature  
S TemporalSequence

    public init S _ S Int where S  
AsyncSequence S Element TemporalFeature Feature

    /// Creates the asynchronous iterator that produces elements of this  
    /// asynchronous sequence.  
    ///  
    /// - Returns: An instance of the `AsyncIterator` type used to  
produce  
    /// elements of the asynchronous sequence.  
public func makeAsyncIterator  
AnyTemporalIterator AnyTemporalSequence Feature Element
```

```

/// Applies each transformer randomly given a probability.
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
public struct ApplyEachRandomly<Element> : RandomTransformer {
    /// The probability of applying each transformer. Default value is 0.5.
    public let probability: Double

    /// Creates an augmentation that applies each transformer randomly in the
    given order.
    ///
    /// - Parameters:
    ///   - probability: The probability of applying each transformer.
    ///     Default value is 0.5.
    ///   - augmentation: An augmentation builder.
    public init(RandomTransformer: Double = 0.5,
                AugmentationBuilder<Element> augmentation) {
        self.probability = RandomTransformer
        self.augmentation = augmentation
    }

    /// Applies each transformer randomly in order with a probability.
    /// - Parameters:
    ///   - input: The input to the transformer.
    ///   - generator: A random number generator.
    ///   - eventHandler: An event handler.
    /// - Returns: The augmented input.
    public func applied(Element inout some RandomNumberGenerator,
                        EventHandler nil async throws Element) {
        let generator = RandomNumberGenerator(some: RandomNumberGenerator)
        let eventHandler = EventHandler(nil)
        let augmentedInput = await augmentation(input, generator, eventHandler)
        return augmentedInput
    }

    /// The input type.
    @available(iOS 17.0, tvOS 17.0, watchOS 11.0, macOS 14.0)
    public typealias Input = Element

    /// The output type.
    @available(iOS 17.0, tvOS 17.0, watchOS 11.0, macOS 14.0)
    public typealias Output = Element
}

/// Randomly applies the transformer with the given probability.
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
public struct ApplyRandomly<RandomTransformer> : RandomTransformer {
    RandomTransformer where RandomTransformer : RandomTransformer
    RandomTransformer Input RandomTransformer Output

    /// The probability of applying the transformer. Default value is 0.5.
    public let probability: Double
}

```

```

    /// Creates an apply randomly augmentation.
    ///
    /// - Parameters:
    ///   - probability: The probability of applying the transformation.
    ///     Must be greater than or equal to 0 and less
    ///       than or equal to 1, the default value is 0.5.
    ///   - augmentation: The transformer to apply randomly.
    public init Input Double 0.5
    AugmentationBuilder Input
    RandomTransformer where Input RandomTransformer Input

    /// Randomly applies a transformer on an input.
    /// - Parameters:
    ///   - input: The input.
    ///   - generator: A random number generator.
    ///   - eventHandler: An event handler.
    /// - Returns: The randomly transformed input.
    public func applied RandomTransformer Input
        inout some RandomNumberGenerator
    EventHandler nil async throws RandomTransformer Output

    /// The input type.
    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Input RandomTransformer Input

    /// The output type.
    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Output RandomTransformer Input

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension ApplyRandomly Sendable where RandomTransformer
Sendable

    /// A transformer for audio conversion.
    @available macOS 13.0 iOS 16.0 tvOS 16.0
    public struct AudioConvertingTransformer Transformer
    Sendable

        /// The target audio format for the output buffers. It must have an
        AVAudioPCMFormat as its common format type.
        public let targetFormat AVAudioFormat

        /// Creates an audio conversion transformer to convert the format of the
        buffers.
        /// - Parameter targetFormat: The desired audio format for the
        output buffers.

```

```

    /// - Precondition The `targetFormat` must have an
    AVAudioPCMFormat as its common format type.
public init AVAudioFormat

    /// Performs conversion of the input audio buffer.
    ///
    /// - Parameters:
    ///   - input: The audio buffer that will be converted.
    ///   - eventHandler: An event handler.
    /// - Returns: An output audio buffer by converting the input buffer to the
    `targetFormat`.
public func applied AVAudioPCMBuffer
    EventHandler nil throws AVAudioPCMBuffer

    /// The input type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input AVAudioPCMBuffer

    /// The output type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output AVAudioPCMBuffer

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension AudioConvertingTransformer
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the

```

```

    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

    /// A stream transformer that extracts audio features from audio buffers.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0)
    public struct AudioFeaturePrint : TemporalTransformer, Sendable

        /// The input type.
        public typealias Input = AVAudioPCMBuffer

        /// The output type.
        public typealias Output = MLShapedArray<Float>

        /// The window duration of the extractor.
        ///
        /// The window duration should be greater than or equal to 0.5 seconds and less than or equal to 15.0 seconds.
        public let windowDuration TimeInterval

        /// The overlap factor of the extractor.
        ///
        /// The overlap should be greater than or equal to zero and less than one.
        public let overlapFactor Double

        /// Creates an audio feature print feature extractor.
        /// - Parameters:
        ///   - windowDuration: The window duration in seconds. The window duration should be greater than or equal to 0.5 seconds and less than or equal to 15.0 seconds.
        ///   - overlapFactor: The overlap factor. The overlap should be greater than or equal to zero and less than one.
        public init(windowDuration: TimeInterval, overlapFactor: Double)

        /// Extracts audio features from an sequence of audio buffers
        ///
        /// You can call this method multiple times to process multiple streams.
        ///
        /// - Parameters:
        ///   - input: An async sequence of audio buffers.
        ///   - eventHandler: An event handler.
        /// - Returns: An async sequence of shaped arrays containing extracted features. Each shaped array has a shape of `[512]`.
        public func applied<S>(eventHandler: EventHandler<S>, nil throws AudioFeaturePrint<Feature> where S : TemporalSequence<Feature>, Feature : AVAudioPCMBuffer)

```

```

    /// The output async sequence type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias OutputSequence
    AudioFeaturePrint FeatureSequence

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension AudioFeaturePrint : CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    struct Point: CustomDebugStringConvertible {
        let x: Int, y: Int
        ///
        var debugDescription: String {
            return "(\(x), \(y))"
        }
    }
    ///
    let p = Point(x: 21, y: 30)
    let s = String(reflecting: p)
    print(s)
    // Prints "(21, 30)"

    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
public var debugDescription: String { get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension AudioFeaturePrint

    /// An async sequence of audio buffers.
    public struct FeatureSequence : TemporalSequence

        /// The feature type.
        public typealias Feature = MLShapedArray<Float>

        /// The type of asynchronous iterator that produces elements of this
        /// asynchronous sequence.
        public typealias AsyncIterator
        AudioFeaturePrint FeatureSequence Iterator

```

```
    /// The number of elements in the sequence. For this sequence count  
is always nil.  
    public var count Int get  
  
    /// Constructs an iterator.  
    public func makeAsyncIterator  
AudioFeaturePrint FeatureSequence AsyncIterator  
  
    /// The type of element produced by this asynchronous sequence.  
    @available iOS 16.0 tvOS 16.0 macOS 13.0  
    public typealias Element  
TemporalFeature AudioFeaturePrint FeatureSequence Feature
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0  
extension AudioFeaturePrint FeatureSequence  
  
    /// An async iterator of audio buffers.  
    public struct Iterator : AsyncIteratorProtocol  
  
        /// Asynchronously advances to the next element and returns it, or  
ends the  
        /// sequence if there is no next element.  
        ///  
        /// - Returns: The next element, if it exists, or `nil` to signal the  
end of  
        /// the sequence.  
        public mutating func next() async throws  
TemporalFeature MLShapedArray Float  
  
        @available iOS 16.0 tvOS 16.0 macOS 13.0  
        public typealias Element  
TemporalFeature MLShapedArray Float
```

```
    /// Audio preprocessing errors.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0  
    public enum AudioPreprocessingError : LocalizedError  
Equatable Sendable  
  
        /// An error that indicates that the input and output formats are incompatible  
for creating an audio converter.  
        case incompatibleTargetFormatForConversion  
AVAudioFormat AVAudioFormat  
  
        /// A localized message describing what error occurred.  
        public var errorDescription String get
```

```
    /// Returns a Boolean value indicating whether two values are equal.  
    ///  
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    ///  
    /// - Parameters:  
    ///   - lhs: A value to compare.  
    ///   - rhs: Another value to compare.  
    public static func AudioPreprocessingError  
    AudioPreprocessingError Bool  
  
    @available macOS 13.0 iOS 16.0 tvOS 16.0  
    extension AudioPreprocessingError  
    CustomDebugStringConvertible  
  
    /// A text representation of the error.  
    public var debugDescription String get  
  
    /// An audio file reader.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0  
    public struct AudioReader Transformer Sendable  
  
    /// The audio reader configuration  
    public var configuration AudioReader Configuration  
  
    /// Creates an audio reader.  
    /// - Parameter configuration: The audio reader configuration.  
    public init AudioReader Configuration  
  
    /// Reads an audio file as an async sequence of audio buffers.  
    ///  
    /// - Parameters:  
    ///   - url: An audio file URL.  
    ///   - eventHandler: An event handler.  
    /// - Returns: An async sequence of `AVAudioPCMBuffer`.  
    public func applied URL  
    EventHandler nil throws AudioReader AsyncBuffers  
  
    /// Reads an audio file as an async sequence of audio buffers.  
    ///  
    /// - Parameter url: An audio file URL.  
    /// - Parameter configuration: The configuration for reading buffers.  
    /// - Returns: An async sequence of `AVAudioPCMBuffer`.  
    public static func read URL  
    AudioReader Configuration throws  
    AudioReader AsyncBuffers
```

```

    /// Reads a sequence of files as an array of async sequences of audio
buffers.
    ///
    /// - Parameter files: A sequence of URLs.
    /// - Parameter configuration: The configuration for reading buffers.
    /// - Returns: An array of async sequences of audio buffers.
    public static func read S _           S
AudioReader Configuration           throws
    AudioReader AsyncBuffers where S Sequence S Element
URL

    /// Reads a sequence of annotated files as a lazy sequence of results each
containing an audio buffers or an error.
    ///
    /// - Parameter annotatedFiles: A sequence of annotated URLs.
    /// - Parameter configuration: The configuration for reading buffers.
    /// - Returns: An array of annotated async sequences.
    public static func read S Annotation -
S           AudioReader Configuration           throws
    AnnotatedFeature AudioReader AsyncBuffers Annotation
where S Sequence Annotation Equatable Annotation
Sendable S Element AnnotatedFeature URL Annotation

    /// Reads an async sequence of audio frames captured with a microphone.
    ///
    /// - Parameter configuration: The configuration for reading buffers.
    /// - Returns: An async sequence of `AVAudioPCMBuffer`.
    @available macOS 13.0 iOS 16.0
    @available
    public static func readMicrophone
AudioReader Configuration           async throws
    AudioReader MicrophoneAsyncBuffers

    /// The input type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias Input URL

    /// The output type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias Output AudioReader AsyncBuffers

```

extension AudioReader

```

    /// An async sequence of audio frames.
    ///
    /// This sequence allows iterating through the microphone audio frames.
    @available macOS 13.0 iOS 16.0
    @available

```

```
public struct MicrophoneAsyncBuffers : TemporalSequence, Sendable

    /// The type of asynchronous iterator that produces elements of this
    /// asynchronous sequence.
    public typealias AsyncIterator = AudioReader<MicrophoneAsyncBuffers>.Iterator

    /// The feature type.
    public typealias Feature = AVAudioPCMBuffer

    /// The number of audio buffers. For this sequence count is always nil.
    public var count: Int { get }

    /// Constructs an iterator.
    public func makeAsyncIterator() = AudioReader<MicrophoneAsyncBuffers>.Iterator

    /// The type of element produced by this asynchronous sequence.
    @available(iOS 16.0, macOS 13.0) public typealias Element = TemporalFeature<AudioReader<MicrophoneAsyncBuffers>, Feature>

@available(macOS 13.0, iOS 16.0, tvOS 16.0)
extension AudioReader
```

```
    /// An async sequence of audio buffers read from an audio file.
    ///
    /// This sequence allows iterating through the file only once.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0)
    public struct AsyncBuffers : TemporalSequence

        /// The type of asynchronous iterator that produces elements of this
        /// asynchronous sequence.
        public typealias AsyncIterator = AudioReader<AsyncBuffers>.Iterator

        /// The feature type.
        public typealias Feature = AVAudioPCMBuffer

        /// The audio file URL, used when throwing an error.
        public let url: URL

        /// The number of audio buffers in the file.
        public let count: Int
```

```
    /// Constructs an iterator.
    public func makeAsyncIterator
AudioReader AsyncBuffers Iterator

    /// The type of element produced by this asynchronous sequence.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias Element
TemporalFeature AudioReader AsyncBuffers Feature

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension AudioReader

    /// The configuration of the audio reader.
    @available macOS 13.0 iOS 16.0 tvOS 16.0
    public struct Configuration Sendable

        /// The maximum size of each buffer in frames. The default value is
`4096`.
        public var frameCount Int

        /// Creates an AudioReader Configuration
        public init

        /// Creates an AudioReader Configuration
        ///
        /// - Parameter frameCount: The maximum size of each buffer in
frames.
        public init Int

@available macOS 13.0 iOS 16.0
available
extension AudioReader MicrophoneAsyncBuffers

    /// An async iterator of audio frames.
    @available macOS 13.0 iOS 16.0
    available
    final public class Iterator AsyncIteratorProtocol

        /// Advances to the next element and returns it, or nil if no next element
exists.
        final public func next async throws
TemporalFeature AudioReader MicrophoneAsyncBuffers Feature

        @available iOS 16.0 macOS 13.0
        @available 16.0
        public typealias Element
```

```
TemporalFeature AudioReader MicrophoneAsyncBuffers Feature
```

```
extension AudioReader AsyncBuffers
```

```
    /// An async iterator of audio buffers.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0  
    public struct Iterator : AsyncIteratorProtocol  
  
        /// Asynchronously advances to the next element and returns it, or  
        ends the  
        /// sequence if there is no next element.  
        ///  
        /// - Returns: The next element, if it exists, or `nil` to signal the  
        end of  
        /// the sequence.  
        public mutating func next() async throws  
TemporalFeature AVAudioPCMBuffer  
  
        @available iOS 16.0 tvOS 16.0 macOS 13.0  
        public typealias Element  
TemporalFeature AVAudioPCMBuffer
```

```
/// Audio reader errors.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public enum AudioReaderError : LocalizedError, Equatable, Sendable
```

```
    /// An error that indicates that the microphone authorization status is denied.
```

```
    /// The user has explicitly denied permission for audio capture.
```

```
    case microphoneAuthorizationDenied
```

```
    /// An error that indicates that the microphone authorization status is  
    restricted.
```

```
    /// The user is not allowed to access audio capture devices.
```

```
    case microphoneAuthorizationRestricted
```

```
    /// An error that indicates that no source devices are available.
```

```
    case sourceDeviceNotAvailable
```

```
    /// A localized message describing what error occurred.
```

```
    public var errorDescription : String { get }
```

```
    /// Hashes the essential components of this value by feeding them into the  
    /// given hasher.
```

```
    ///
```

```
    /// Implement this method to conform to the `Hashable` protocol. The
```

```
    /// components used for hashing must be the same as the components  
    compared
```

```
    /// in your type's `==` operator implementation. Call  
    `hasher.combine(_:)`  
    /// with each of these components.
```

```
    ///
```

```
    /// - Important: In your implementation of `hash(into:)`,  
    /// don't call `finalize()` on the `hasher` instance provided,  
    /// or replace it with a different instance.
```

```
    /// Doing so may become a compile-time error in the future.
```

```
    ///
```

```
    /// - Parameter hasher: The hasher to use when combining the  
components
```

```
    /// of this instance.
```

```
public func hash           inout Hasher
```

```
    /// Returns a Boolean value indicating whether two values are equal.
```

```
    ///
```

```
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.
```

```
    ///
```

```
    /// - Parameters:
```

```
    /// - lhs: A value to compare.
```

```
    /// - rhs: Another value to compare.
```

```
public static func           AudioReaderError
```

```
AudioReaderError      Bool
```

```
    /// The hash value.
```

```
    ///
```

```
    /// Hash values are not guaranteed to be equal across different executions of  
    /// your program. Do not save hash values to use during a future execution.
```

```
    ///
```

```
    /// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To
```

```
    /// conform to `Hashable`, implement the `hash(into:)` requirement  
instead.
```

```
    /// The compiler provides an implementation for `hashValue` for you.
```

```
public var hashValue Int   get
```

```
@available macOS 13.0  iOS 16.0  tvOS 16.0
```

```
extension AudioReaderError  CustomDebugStringConvertible
```

```
    /// A text representation of the error.
```

```
public var debugDescription String   get
```

```
@available macOS 13.0  iOS 16.0  tvOS 16.0
```

```
extension AudioReaderError  Hashable
```

```
/// A series of augmentations.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
@resultBuilder public struct AugmentationBuilder Element  
Sendable  
  
    /// Builds a partial result random transformer from the first random  
transformer.  
    /// - Parameter first: A random transformer.  
    public static func buildPartialBlock some  
RandomTransformer Element Element some  
RandomTransformer Element Element  
  
    /// Builds a partial result from the first transformer.  
    /// - Parameter first: A transformer.  
    public static func buildPartialBlock some  
Transformer Element Element some  
RandomTransformer Element Element  
  
    /// Builds a partial result by combining an accumulated random transformer  
and a new random transformer.  
    /// - Parameters:  
    ///   - accumulated: A random transformer representing the  
accumulated result thus far.  
    ///   - next: A random transformer representing the next component  
after the accumulated ones in the block.  
    public static func buildPartialBlock some  
RandomTransformer Element Element some  
RandomTransformer Element Element some  
RandomTransformer Element Element  
  
    /// Builds a partial result by combining an accumulated random transformer  
and a new transformer.  
    /// - Parameters:  
    ///   - accumulated: A random transformer representing the  
accumulated result thus far.  
    ///   - next: A transformer representing the next component after the  
accumulated ones in the block.  
    public static func buildPartialBlock some  
RandomTransformer Element Element some  
Transformer Element Element some  
RandomTransformer Element Element
```

```

    /// An async sequence of augmented elements.
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
public struct AugmentationSequence<Base: RandomTransformer<RandomNumberGenerator, Annotation, AsyncSequence>, Where: Sequence, RandomTransformer: RandomTransformer<RandomNumberGenerator, RandomNumberGenerator, Base, Element>, AnnotatedFeature: RandomTransformer<Input, Annotation>, RandomTransformer: RandomTransformer<Input, RandomTransformer<Output>>
{
    /// The type of element produced by this asynchronous sequence.
    public typealias Element = Base.Element

    /// The transformation applied to each element.
    public var transformer: RandomTransformer<RandomNumberGenerator, Annotation, AsyncIterator<Element>>

    /// Creates the asynchronous iterator that produces elements of this
    /// asynchronous sequence.
    public func makeAsyncIterator() async returns (AsyncIterator<Element>)
}

extension AugmentationSequence<Base: RandomTransformer<RandomNumberGenerator, Annotation, AsyncSequence>, Where: Sequence, RandomTransformer: RandomTransformer<RandomNumberGenerator, RandomNumberGenerator, Base, Element>, AnnotatedFeature: RandomTransformer<Input, Annotation>, RandomTransformer: RandomTransformer<Input, RandomTransformer<Output>> {
    /// The iterator that produces elements in the augmentation sequence.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
    public struct AsyncIterator<T>: AsyncIteratorProtocol {
        /// Produces the next element in the augmentation sequence.
        public mutating func next() async throws returns (T)
    }

    /// Available on macOS 14.0 and later.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0, macOS)
    public typealias Element = Base.Element
}

extension AugmentationSequence<Base: RandomTransformer<RandomNumberGenerator, Annotation, AsyncSequence>, Where: Sequence, RandomTransformer: RandomTransformer<RandomNumberGenerator, RandomNumberGenerator, Base, Element>, AnnotatedFeature: RandomTransformer<Input, Annotation>, RandomTransformer: RandomTransformer<Input, RandomTransformer<Output>> {
    /// Batches a augmentation sequence.
    ///
    /// - Parameters:
    ///   - size: The number of elements contained in each batch.
    ///   - dropsLastPartialBatch: A Boolean value representing whether the last batch should be dropped if it has less
    ///     than `size` elements.
    /// - returns: An async sequence of batches.
}

```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public func batches Int
Bool AugmentationSequence Base
RandomTransformer RandomNumberGenerator
Annotation BatchedSequence

/// An async sequence that batches an augmentation sequence.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public struct BatchedSequence AsyncSequence

/// The type of element produced by this asynchronous sequence.
public typealias Element AugmentationSequence Base
RandomTransformer RandomNumberGenerator Annotation Element

/// Creates the asynchronous iterator that produces batches.
public func makeAsyncIterator
AugmentationSequence Base RandomTransformer
RandomNumberGenerator
Annotation BatchedSequence AsyncIterator

extension AugmentationSequence BatchedSequence

/// The iterator that produces batches.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public struct AsyncIterator AsyncIteratorProtocol

/// Produces the next batch.
public mutating func next async throws
AugmentationSequence Base RandomTransformer
RandomNumberGenerator Annotation Element

@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0
public typealias Element AugmentationSequence Base
RandomTransformer RandomNumberGenerator Annotation Element

/// An augmenter.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct Augmener RandomTransformer
RandomNumberGenerator where RandomTransformer
RandomTransformer RandomNumberGenerator
RandomNumberGenerator RandomTransformer Input
```

RandomTransformer Output

```
    /// Creates an augmenter from a random number generator and an
    augmentation builder.
    ///
    /// - Parameters:
    ///   - generator: A random number generator.
    ///   - builder: An augmentation builder.
    public init Input RandomNumberGenerator
                    AugmentationBuilder Input
                    RandomTransformer where Input
                    RandomTransformer Input
                    ->
                    RandomTransformer Input

    /// Applies an augmentation per input of the base sequence.
    ///
    /// - Parameter base: A sequence of elements to augment.
    /// - Returns: A sequence of augmented elements having the same
    number of elements as the input sequence.
    public func applied S Annotation S
    AugmentationSequence S RandomTransformer
    RandomNumberGenerator Annotation where S Sequence
    Annotation Equatable S Element
    AnnotatedFeature RandomTransformer Input Annotation

    /// Applies an augmentation repeatedly to an array of inputs.
    ///
    /// - Parameters:
    ///   - elements: A collection of elements to augment.
    ///   - count: The number of times to shuffle and augment the input
    elements. Must be at least one.
    /// - Returns: A sequence of augmented elements having `count`
    times the number of elements in the input collection.
    public func applied C Annotation C
                    Int UpsampledAugmentationSequence C
    RandomTransformer RandomNumberGenerator Annotation where
    C Collection Annotation Equatable C Element
    AnnotatedFeature RandomTransformer Input Annotation

    /// A gradient boosted decision tree classifier.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public struct BoostedTreeClassifier Label
    SupervisedTabularEstimator where Label Comparable Label
    Decodable Label Encodable Label Hashable

    /// The transformer type created by this estimator.
    public typealias Transformer TreeClassifierModel Label

    /// The annotation type.
    public typealias Annotation Label
```

```

    /// Boosted tree configuration.
public var configuration BoostedTreeConfiguration

    /// The annotation column identifier.
public var annotationColumnID ColumnID Label

    /// The names of the columns containing feature values.
public var featureColumnNames String

    /// The set of possible labels.
public var labels Set Label

    /// Creates a boosted tree classifier.
    ///
    /// - Parameters
    ///   - labels: The set of possible labels.
    ///   - annotationColumnName: The name of the column containing the
    ground truth labels.
    ///   - featureColumnNames: The names of the feature columns.
    ///   - configuration: The configuration.
public init Set Label
String String
BoostedTreeConfiguration

    /// Fits a boosted tree classifier model to a collection of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame of examples.
    ///   - validation: A data frame of validation examples.
    ///   - eventHandler: An event handler. This method reports accuracy
and loss metrics.
    /// - Returns: The fitted boosted tree classifier model.
public func fitted DataFrame
DataFrame nil EventHandler
nil async throws TreeClassifierModel Label

    public func encodeLabels _ some
Collection Label throws String
Int

    /// Encodes a fitted transformer.
public func encode _ inout any
TreeClassifierModel Label inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
public func decode inout any
EstimatorDecoder throws TreeClassifierModel Label

```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension BoostedTreeClassifier
UpdatableSupervisedTabularEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func makeTransformer
TreeClassifierModel Label

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///     - transformer: A transformer to update.
    ///     - input: A data frame containing examples.
    ///     - eventHandler: An event handler.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func update _ inout
TreeClassifierModel Label DataFrame
EventHandler async throws

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///     - transformer: A transformer this estimator creates.
    ///     - encoder: An encoder.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func encodeWithOptimizer _ inout any
TreeClassifierModel Label
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func decodeWithOptimizer inout any
EstimatorDecoder throws TreeClassifierModel Label

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension BoostedTreeClassifier Sendable where Label
Sendable
```

```

/// A boosted tree configuration.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct BoostedTreeConfiguration : Hashable, Codable, Sendable

    /// The learning rate.
    ///
    /// The learning rate controls the step size shrinkage. A smaller learning rate makes the learning process more
    /// conservative. Must be in the range [0, 1]. Defaults to 0.3.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public var learningRate : Double

    /// Maximum tree depth.
    ///
    /// For best results use a value between 4 and 8. Must be at least 1. Defaults to 6.
    public var maximumDepth : Int

    /// Maximum number of iterations.
    public var maximumIterations : Int

    /// Minimum loss reduction required to further split a node during the tree learning phase.
    ///
    /// Larger values can help prevent overfitting by avoiding splits that do not sufficiently reduce the loss function.
    /// Defaults to 0.
    public var minimumLossReduction : Double

    /// The minimum weight of each leaf node.
    ///
    /// Larger values result in more conservative tree learning and help prevent overfitting. If the tree learning
    /// algorithm results in a leaf node with the sum of instance weights less than this value, tree building will
    /// terminate. Defaults to 0.1
    public var minimumChildWeight : Double

    /// A seed to generate reproducible results from random operations such as column and row subsampling.
    public var randomSeed : Int

    /// The step size shrinking.
    @available(macOS 13.0, iOS 14.0)
    "learningRate"
    @available(macOS 16.0, iOS 17.0)
    "learningRate"
    @available(macOS 16.0, iOS 17.0)

```

```
        "learningRate"
public var stepSize Double

    /// Stops training after this number of iterations where the validation metric
does not improve.
    ///
    /// Validation data must be specified for an early stop.
public var earlyStoppingIterationCount Int

    /// Subsample ratio of the training set in each iteration of tree construction.
    ///
    /// This is called the bagging trick and can usually help prevent overfitting.
Setting this to a value of 0.5
    /// results in the model randomly sampling half of the examples (rows) to
grow each tree. Must be in the range
    /// (0, 1).
public var rowSubsample Double

    /// Subsample ratio of the columns in each iteration of tree construction.
    ///
    /// Like row subsample, this can also help prevent model overfitting. Setting
this to a value of 0.5 results in the
    /// model randomly sampling half of the columns to grow each tree. Must be
in the range (0, 1).
public var columnSubsample Double

    /// The number of parallel trees constructed during each iteration.
    ///
    /// Use a value greater than one to train a boosted random forest.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var parallelTreeCount Int

    /// Creates a default boosted tree configuration.
public init

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,
    /// don't call `finalize()` on the `hasher` instance provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
```

```
///  
/// - Parameter hasher: The hasher to use when combining the  
components  
/// of this instance.  
public func hash inout Hasher  
  
/// Returns a Boolean value indicating whether two values are equal.  
///  
/// Equality is the inverse of inequality. For any values `a` and `b`,  
/// `a == b` implies that `a != b` is `false`.  
///  
/// - Parameters:  
/// - lhs: A value to compare.  
/// - rhs: Another value to compare.  
public static func BoostedTreeConfiguration  
BoostedTreeConfiguration Bool  
  
/// Encodes this value into the given encoder.  
///  
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.  
///  
/// This function throws an error if any values are invalid for the given  
/// encoder's format.  
///  
/// - Parameter encoder: The encoder to write data to.  
public func encode any Encoder throws  
  
/// The hash value.  
///  
/// Hash values are not guaranteed to be equal across different executions of  
/// your program. Do not save hash values to use during a future execution.  
///  
/// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To  
/// conform to `Hashable`, implement the `hash(into:)` requirement  
instead.  
/// The compiler provides an implementation for `hashValue` for you.  
public var hashValue Int get  
  
/// Creates a new instance by decoding from the given decoder.  
///  
/// This initializer throws an error if reading from the decoder fails, or  
/// if the data read is corrupted or otherwise invalid.  
///  
/// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws  
  
/// A gradient boosted decision tree regressor.
```

```

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct BoostedTreeRegressor Annotation
SupervisedTabularEstimator

    /// The transformer type created by this estimator.
    public typealias Transformer TreeRegressorModel

    /// Boosted tree configuration.
    public var configuration BoostedTreeConfiguration

    /// The annotation column identifier.
    public var annotationColumnID ColumnID Annotation

    /// The names of the columns containing feature values.
    public var featureColumnNames String

    /// Creates a boosted tree regressor.
    ///
    /// - Parameters
    ///   - annotationColumnName: The name of the column containing the ground truth values.
    ///   - featureColumnNames: The names of the feature columns.
    ///   - configuration: The configuration.
    public init String
        String
    BoostedTreeConfiguration

    /// Fits a boosted tree regressor model to a collection of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples used for fitting the transformer.
    ///   - validation: A data frame containing examples used for validating the fitted transformer.
    ///   - eventHandler: An event handler. This method reports maximum error and root-mean-square error metrics.
    /// - Returns: The fitted boosted tree regressor model.
    public func fitted DataFrame
        DataFrame nil
        EventHandler
    nil async throws TreeRegressorModel

    /// Encodes a fitted transformer.
    public func encode _ TreeRegressorModel
        inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode inout any
    EstimatorDecoder throws TreeRegressorModel

```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension BoostedTreeRegressor
UpdatableSupervisedTabularEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func makeTransformer TreeRegressorModel

    /// Updates a transformer with a new sequence of examples.
< bbb/>
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A data frame containing examples.
    ///   - eventHandler: An event handler.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func update _ inout
TreeRegressorModel DataFrame
EventHandler async throws

    /// Encodes the transformer and optimizer to an encoder.
< bbb/>
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func encodeWithOptimizer _ inout any EstimatorEncoder
TreeRegressorModel
throws

    /// Reads the encoded transformer and optimizer with a decoder.
< bbb/>
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func decodeWithOptimizer _ inout any
EstimatorDecoder
throws TreeRegressorModel

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension BoostedTreeRegressor Sendable where Annotation
Sendable

    /// An estimator that replaces missing values in the categorical input.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public struct CategoricalImputer<Element> : Estimator where
    Element : Decodable, Element : Encodable, Element : Hashable

    /// The transformer type created by this estimator.
    public typealias Transformer = ImputeTransformer<Element>

    /// An imputation strategy.
    public enum Strategy {
        /// Imputation strategy that replaces missing elements with the mode.
        case mode

        /// Imputation strategy that replaces missing elements with a constant.
        case constant(Element)
    }

    /// The imputation strategy.
    public var strategy: CategoricalImputer<Element>.Strategy

    /// Creates an imputer with a strategy.
    public init(_ strategy: CategoricalImputer<Element>.Strategy)

    /// Creates an imputer with a constant value to use when replacing missing
    values.
    public init(Element constantValue)

    /// Fits a categorical imputer to a sequence of elements.
    ///
    /// - Parameters:
    ///   - input: A sequence of elements.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted<S>(EventHandler<nil> eventHandler) -> S
        where S : Sequence, S.Element == Element, Transformer where S.Element == Element

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension CategoricalImputer<Element> : CustomDebugStringConvertible
```

```
    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
```

```

////
////    struct Point: CustomDebugStringConvertible {
////        let x: Int, y: Int
////
////        var debugDescription: String {
////            return "(\(x), \(y))"
////        }
////    }
////
////    let p = Point(x: 21, y: 30)
////    let s = String(reflecting: p)
////    print(s)
////    // Prints "(21, 30)"
////
////    /// The conversion of `p` to a string in the assignment to `s` uses the
////    /// `Point` type's `debugDescription` property.
public var debugDescription String get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension CategoricalImputer Sendable where Element
Sendable

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension CategoricalImputer Strategy Sendable where Element
Sendable

```

```

/// Apply single transformation randomly chosen from a list of transformers.

```

```

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct ChooseRandomly Element RandomTransformer

```

```

/// Creates a choose randomly augmentation.
///
/// - Parameters:
///   - augmentation: An augmentation builder.
public
init RandomTransformer AugmentationBuilder Element -
    RandomTransformer where Element =
RandomTransformer Input RandomTransformer
RandomTransformer RandomTransformer Input
RandomTransformer Output

```

```

/// Chooses a random transformer from a list of transformers and applies the
chosen transformer.

```

```

/// - Parameters:
///   - input: The input to the transformer.
///   - generator: A random number generator.

```

```
/// - eventHandler: An event handler.
/// - Returns: The augmented input.
public func applied Element inout
some RandomNumberGenerator EventHandler nil
async throws Element

/// The input type.
@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

public typealias Input Element

/// The output type.
@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

public typealias Output Element

/// An item in a classification result.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct Classification Label Hashable where Label Hashable

/// The classification label.
public var label Label

/// The classification probability. A value between 0 and 1.
public var probability Float

/// Creates a classification with label and probability.
///
/// - Parameters
///   - label: The classification label.
///   - probability: The classification probability.
public init Label Float

/// Hashes the essential components of this value by feeding them into the
/// given hasher.
///
/// Implement this method to conform to the `Hashable` protocol. The
/// components used for hashing must be the same as the components
compared
/// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
/// with each of these components.
///
/// - Important: In your implementation of `hash(into:)`,
/// don't call `finalize()` on the `hasher` instance provided,
/// or replace it with a different instance.
/// Doing so may become a compile-time error in the future.
```

```

    /**
     * - Parameter hasher: The hasher to use when combining the
     components
     *   of this instance.
public func hash           inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    /**
     * Equality is the inverse of inequality. For any values `a` and `b`,
     * `a == b` implies that `a != b` is `false`.
    /**
     * - Parameters:
     *   - lhs: A value to compare.
     *   - rhs: Another value to compare.
public static func      Classification Label
Classification Label      Bool

    /// The hash value.
    /**
     * Hash values are not guaranteed to be equal across different executions of
     * your program. Do not save hash values to use during a future execution.
    /**
     * - Important: `hashValue` is deprecated as a `Hashable` requirement. To
     * conform to `Hashable`, implement the `hash(into:)` requirement
     * instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Classification Codable where Label Decodable
Label Encodable

    /// Encodes this value into the given encoder.
    /**
     * If the value fails to encode anything, `encoder` will encode an empty
     * keyed container in its place.
    /**
     * This function throws an error if any values are invalid for the given
     * encoder's format.
    /**
     * - Parameter encoder: The encoder to write data to.
public func encode      any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    /**
     * This initializer throws an error if reading from the decoder fails, or
     * if the data read is corrupted or otherwise invalid.
    /**

```

```
/// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Classification Sendable where Label Sendable

/// A classification distribution that contains a probability for each classification
label.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct ClassificationDistribution Label Collection
where Label Hashable

/// The label with the highest probability.
public var mostLikelyLabel Label get

/// The labels sorted by decreasing probability.
public var labelsSortedByProbability Label get

/// The index of the initial element in the classification distribution.
public var startIndex Int get

/// The index of the final element in the classification distribution.
public var endIndex Int get

/// Creates a classification distribution.
///
/// - Parameter classifications: A collection of classifications.
/// - Precondition: The classifications must contain unique labels.
public init C where C Collection
C Element Classification Label

/// Computes the most likely labels in the classification set.
///
/// - Parameter amount: The number of top labels.
/// - Returns: The labels with the highest probabilities.
public func topLabels _ Int Label

/// Accesses a classification at an index.
///
/// - Parameter index: A valid index to a classification in the
classification distribution.
public subscript Int Classification Label
get

/// Accesses a probability with label.
///
/// - Parameter label: A label in the classification set.
public subscript Label Float get
```

```
/// Accesses a contiguous range of elements.  
///  
/// - Parameter bounds: A range of valid indices in the classification  
distribution.  
public subscript Range Int  
Slice ClassificationDistribution Label get  
  
/// Returns the index immediately after an element index.  
/// - Parameter i: A valid index to an element in the classification  
distribution.  
public func index Int Int  
  
/// Returns the index immediately before an element index.  
/// - Parameter i: A valid index to an element in the classification  
distribution.  
public func index Int Int  
  
/// Creates a new classification distribution by applying a transformation to  
every element.  
///  
/// - Parameter transform: A transformation closure.  
public func map T _ Classification Label  
throws Classification T rethrows  
ClassificationDistribution T where T Hashable  
  
/// A type representing the sequence's elements.  
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
public typealias Element Classification Label  
  
/// A type that represents a position in the collection.  
///  
/// Valid indices consist of the position of every element and a  
/// "past the end" position that's not valid for use as a subscript  
/// argument.  
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
public typealias Index Int  
  
/// A type that represents the indices that are valid for subscripting the  
/// collection, in ascending order.  
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
public typealias Indices  
DefaultIndices ClassificationDistribution Label  
  
/// A type that provides the collection's iteration interface and  
/// encapsulates its iteration state.  
///
```

```
/// By default, a collection conforms to the `Sequence` protocol by
/// supplying `IndexingIterator` as its associated `Iterator`
/// type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Iterator
IndexingIterator ClassificationDistribution Label

/// A collection representing a contiguous subrange of this collection's
/// elements. The subsequence shares indices with the original collection.
///
/// The default subsequence type for collections that don't define their own
/// is `Slice`.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias SubSequence
Slice ClassificationDistribution Label

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ClassificationDistribution : Equatable

/// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:
///   - lhs: A value to compare.
///   - rhs: Another value to compare.
public static func
ClassificationDistribution Label
ClassificationDistribution Label Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ClassificationDistribution : Hashable

/// Hashes the essential components of this value by feeding them into the
/// given hasher.
///
/// Implement this method to conform to the `Hashable` protocol. The
/// components used for hashing must be the same as the components
/// compared
/// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
/// with each of these components.
///
/// - Important: In your implementation of `hash(into:)`,
```

```

    //// don't call `finalize()` on the `hasher` instance provided,
    //// or replace it with a different instance.
    //// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash           inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable` requirement. To
conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ClassificationDistribution Codable where Label
Decodable Label Encodable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode           any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init           any Decoder throws

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ClassificationDistribution Sendable where Label
Sendable

```

```

/// Classification metrics.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct ClassificationMetrics<Label> where Label: Hashable

    /// The number of examples used to compute the metrics.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public var exampleCount: Int { get }

    /// The set of labels.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public var labels: Set<Label>

    /// A Boolean value indicating whether to restrict metrics to labels in the
    labels set.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public var restrictToKnownLabels: Bool

    /// The number of correctly classified examples out of the total number of
    examples.
    public var accuracy: Double

    /// Creates empty classification metrics.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public init()

    /// Creates classification metrics for predicted and ground truth labels.
    ///
    /// - The predicted and ground truth collections are matched element by
    element in the order they are provided. Both
    /// collections must have the same number of elements. Labels not in the
    labels set are ignored.
    ///
    /// - Parameters
    ///   - predicted: The predicted labels.
    ///   - groundTruth: The true labels.
    ///   - labels: The set of labels to consider.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public init<Predicted, Correct, PredictedElement, PredictedSequence, PredictedCorrectElement, PredictedSetLabel, PredictedCorrectElement, PredictedLabel>(predicted: Predicted, groundTruth: Correct, labels: PredictedSetLabel) where Predicted: Sequence, PredictedElement: Element, PredictedSequence: Sequence, PredictedCorrectElement: Correct, PredictedLabel: Label

    /// Creates classification metrics for predicted and ground truth labels.

```

```

    /**
     * The predicted and ground truth collections are matched element by
     * element in the order they are provided. Both
     *   collections must have the same number of elements.
     */
    /**
     * - Parameters
     *   - predicted: The predicted labels.
     *   - groundTruth: The true labels.
public init Predicted Correct - Predicted -
              Correct where Label Predicted Element
Predicted Collection Correct Collection
Predicted Element   Correct Element

    /**
     * Creates classification metrics for a sequence of predicted and ground
     * truth label pairs.
     */
    /**
     * - Parameters:
     *   - pairs: A sequence of predicted and true label pairs.
     *   - labels: The set of labels to consider.
available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public init _ some Sequence Label
          Label Set Label

    /**
     * Creates classification metrics for a sequence of predicted and ground
     * truth label pairs.
     */
    /**
     * - Parameters:
     *   - pairs: A sequence of predicted and true label pairs.
available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public init _ some Sequence Label
          Label

    /**
     * Creates classification metrics from a temporal sequence of annotated
     * classifications.
     */
    /**
     * - Parameters
     *   - predicted: The predicted sequence of annotated temporal
     *     sequences of classification distributions.
public init S Inner - S async throws where S
Sequence Inner TemporalSequence S Element
AnnotatedFeature Inner Label Inner Feature
ClassificationDistribution Label

    /**
     * Updates the metrics with more predicted and ground truth labels.
     */
    /**
     * The predicted and ground truth sequences are matched element by
     * element in the order they are provided. Both
     *   sequences must have the same number of elements.

```

```
///  
/// - Parameters:  
///   - predicted: The predicted labels.  
///   - groundTruth: The true labels.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public mutating func add some Sequence Label  
    some Sequence Label  
  
/// Updates the metrics with more predicted and ground truth label pairs.  
///  
/// - Parameter pairs: A collection of predicted and true label pairs.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public mutating func add _ some  
Sequence Label _ some  
    Label  
  
/// Returns the number of times a label appeared in the ground truth  
collection.  
///  
/// - Parameter label: The label.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func count Label Int  
  
/// Returns the number of times a label appeared in the predicted collection.  
///  
/// - Parameter predicted: The label.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func count Label Int  
  
/// Returns the number of times a predicted, true label pair appeared in the  
label collections.  
///  
/// - Parameters:  
///   - predicted: The predicted label.  
///   - label: The true label.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func count Label Label Int  
  
/// Returns the number of times the predicted label matched the true label.  
///  
/// - Parameter label: A label.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func truePositiveCount Label Int  
  
/// Returns the number of times the predicted label did not match the true
```

```

label.

    /**
     * - Parameter label: A label.
     @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func falsePositiveCount           Label      Int
        /// Returns the number of times a label was not in the predicted or ground
        truth collections.

    /**
     * - Parameter label: A label.
     @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func trueNegativeCount          Label      Int
        /// Returns the number of times a true label was not predicted.

    /**
     * - Parameter label: A label.
     @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func falseNegativeCount         Label      Int
        /// Computes the confusion matrix.

    /**
     * The `i`th row and `j`th column value indicate the count of true label
     being the `i`th class and predicted
     label being the `j`th class. The labels are sorted in ascending order.
    public func makeConfusionMatrix       MLShapedArray<Float>
    where Label : Comparable, Label : Decodable, Label : Encodable

        /// Returns new classification metrics where the labels are the result of
        applying a transformation.

    /**
     * The transformation can combine separate labels into one. The metrics will
     be adjusted accordingly by combining
     counts from the original labels. An example of this is combining mixed
     cased labels into lowercase:
    /**
     * swift
     let metrics = ClassificationMetrics(predicted,
     groundTruth).map({ $0.lowercased() })
    /**
     @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func mapLabels<T>           Label      throws T
    rethrows ClassificationMetrics<T> where T : Hashable
        @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
        extension ClassificationMetrics<T>

```

```

    /// Computes the precision score for a class label.
    ///
    /// Precision score is computed as the ratio `tp / (tp + fp)` where
`tp` is the number of true positives and `fp`
    /// is the number of false positives.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The precision score for the given label.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func precisionScore Label Double

    /// Computes the recall score for a class label.
    ///
    /// Precision score is computed as the ratio `tp / (tp + fn)` where
`tp` is the number of true positives and `fn`
    /// is the number of false negatives.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The recall score for the given label.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func recallScore Label Double

    /// Computes the F1 score for a class label.
    ///
    /// The balanced F-score, or F1 score, is computed as the harmonic mean of
the precision and recall.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The F1 score for the given label.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

```

public func f1Score **Label** **Double**

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ClassificationMetrics Sendable where Label
Sendable

```

```

    /// An estimator that predicts classification probabilities.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol Classifier Transformer

    /// The classification label type.
    associatedtype Label Hashable where Self Output
ClassificationDistribution Self Label

```

```
/// A transformer that concatenates every numerical column in a dataframe into to  
a shaped array for each row.
```

```
///
```

```
/// The resulting concatenated column contains `MLShapedArray<Scalar>`  
elements. For example
```

```
///
```

```
///
```

		label	price	rooms	A	B	C
		<String>	<Int>	<Int>	<Int>	<Int>	
		<Int>					
0	0	good	850,000	4	1	0	
0	1	bad	700,000	3	0	1	
1	2	bad	650,000	3	0	0	
0	3	good	600,000	2	0	1	

```
///
```

```
/// would be concatenated as:
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
///
```

```
    /// The selection of columns to concatenate.
    public var columnSelection ColumnSelection

    /// The name of the concatenated column containing the shaped arrays.
    public var concatenatedColumnName String

    /// Creates a concatenator that concatenates numeric columns into a new
    column of ML shaped array.
    ///
    /// - Parameters:
    ///   - columnSelection: A selection of columns to concatenate.
    ///   - concatenatedColumnName: The name of the resulting shaped
    array column.
    public init ColumnSelection
        String "features"

    /// Combines every numerical column in a data frame into to a shaped array
    for each row.
    public func applied DataFrame
    EventHandler nil throws DataFrame

    /// The input type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Input DataFrame

    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Output DataFrame

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ColumnConcatenator CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`-
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
```

```

////
////    let p = Point(x: 21, y: 30)
////    let s = String(reflecting: p)
////    print(s)
////    // Prints "(21, 30)"
////
//// The conversion of `p` to a string in the assignment to `s` uses the
//// `Point` type's `debugDescription` property.
public var debugDescription String get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ColumnConcatenator Sendable where Scalar
Sendable

/// A selection of columns from a data frame.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public enum ColumnSelection Codable Sendable

/// Select all columns in the data frame.
case all

/// Select all numeric columns in the data frame.
///
/// Numeric columns are columns with elements of type `Int`, `UInt8`,
`Float`, `Double`. Also arrays of those
/// types and shaped arrays of those types.
case numeric

/// Selects only the specified columns.
case include String

/// Selects all columns except the specified columns.
case exclude String

/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

/// Creates a new instance by decoding from the given decoder.
///

```

```

    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
    public init any Decoder throws

    /// An operation that applies an estimator to a selection of columns.
    ///
    /// This estimator applies a non-tabular estimator to a selection of columns. Here's
    /// an example of normalizing
    /// numeric values within each column using a ``StandardScaler``:
    ///
    ///     let numericalScaling = ColumnSelector(
    ///         columns: ["volume", "price"],
    ///         estimator: NumericImputer<Float>(.mean)
    ///             .appending(StandardScaler<Float>())
    ///     )
    ///
    /// In most cases, an inputer must handle missing values.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct ColumnSelector Estimator UnwrappedInput
TabularEstimator where Estimator Estimator
Estimator Transformer Input UnwrappedInput

    /// The column selection strategy.
    public var columnSelection ColumnSelection

    /// The estimator to use on each column.
    public var estimator Estimator

    /// The transformer type created by this estimator.
    public typealias Transformer
ColumnSelectorTransformer Estimator Transformer
UnwrappedInput

    public typealias Input Estimator Transformer Input

    public typealias Output Estimator Transformer Output

    /// Creates a select operation with an estimator.
    ///
    /// - Parameters:
    ///   - columnSelection: A selection of columns.
    ///   - estimator: An estimator.
    public init _ ColumnSelection
Estimator

    /// Creates a select operation with a transformer.

```

```

    /**
     * - Parameters:
     *   - columnSelection: A selection of columns.
     *   - transformer: A transformer.
     */
    public init T _ ColumnSelection
        T where Estimator
    TransformerToEstimatorAdaptor T T Transformer T Input
    UnwrappedInput

    /**
     * Creates a select operation with an estimator.
     */
    /**
     * - Parameters:
     *   - columns: An array of columns.
     *   - estimator: An estimator.
     */
    public init String Estimator

    /**
     * Fits a transformer to a data frame
     */
    /**
     * - Parameters:
     *   - input: A data frame.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     */
    public func fitted DataFrame
    EventHandler nil async throws ColumnSelector Estimator
    UnwrappedInput Transformer

    /**
     * Encodes a fitted transformer.
     */
    public func encode _ ColumnSelector Estimator UnwrappedInput Transformer
        inout any EstimatorEncoder throws

    /**
     * Decodes a previously fitted transformer.
     */
    public func decode inout any
    EstimatorDecoder throws ColumnSelector Estimator
    UnwrappedInput Transformer

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension ColumnSelector UpdatableTabularEstimator where
Estimator UpdatableEstimator

    /**
     * Creates a default-initialized transformer suitable for incremental fitting.
     */
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
    public func makeTransformer
    ColumnSelectorTransformer Estimator Transformer
    UnwrappedInput

    /**
     * Updates a transformer with a new sequence of examples.
     */

```

```

    /**
     * - Parameters:
     *   - transformer: A transformer to update.
     *   - input: A sequence of examples.
     *   - eventHandler: An event handler.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
    public func update _ inout
        ColumnSelector Estimator UnwrappedInput Transformer
            DataFrame EventHandler nil async
    throws

    /**
     * Encodes the transformer and optimizer to an encoder.
     */
    /**
     * - Parameters:
     *   - transformer: A transformer this estimator creates.
     *   - encoder: An encoder.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
    public func encodeWithOptimizer _ inout any EstimatorEncoder throws

    /**
     * Reads the encoded transformer and optimizer with a decoder.
     */
    /**
     * - Parameter decoder: A decoder.
     * - Returns: The decoded transformer.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
    public func decodeWithOptimizer inout any
        EstimatorDecoder throws ColumnSelector Estimator
            UnwrappedInput Transformer

    /**
     * A transformer that applies a base transformer to specific columns in a data frame.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    extension ColumnSelector Sendable where Estimator: Sendable, UnwrappedInput: Sendable

    /**
     * A transformer that applies a base transformer to specific columns in a data frame.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    public struct ColumnSelectorTransformer<Base: Transformer> {
        let base: Base
        let unwrappedInput: UnwrappedInput
        let tabularTransformer: TabularTransformer<where Base: Transformer>
        let input: Input<UnwrappedInput>
    }

    /**
     * A dictionary of column names to transformers.
     */
    public var transformers: [String: Base]

```

```
    /// A mapping of input column names to output column names.
    public var columnMapping String String

    /// Creates a select transformer.
    /// - Parameters:
    ///   - transformers: A dictionary of column names to transformers.
    ///   - columnMapping: A mapping of input column names to output
    column names.
    public init String Base
    String String

    /// Performs the transformation on selected columns of the data frame.
    ///
    /// - Parameters:
    ///   - input: A data frame.
    ///   - eventHandler: An event handler.
    /// - Returns: A data frame produced by applying the transformer to
    selected columns.
    public func applied DataFrame
    EventHandler nil async throws DataFrame

    /// The input type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Input DataFrame

    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Output DataFrame

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ColumnSelectorTransformer
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`-
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
```

```
    /**
     */
    let p = Point(x: 21, y: 30)
    let s = String(reflecting: p)
    print(s)
    // Prints "(21, 30)"

    /**
     * The conversion of `p` to a string in the assignment to `s` uses the
     * `Point` type's `debugDescription` property.
    */
public var debugDescription String get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ColumnSelectorTransformer Sendable where Base
Sendable UnwrappedInput Sendable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ColumnSelectorTransformer Codable where Base
Decodable Base Encodable
```

```
    /**
     * Encodes this value into the given encoder.
     */
    /**
     * If the value fails to encode anything, `encoder` will encode an empty
     * keyed container in its place.
     */
    /**
     * This function throws an error if any values are invalid for the given
     * encoder's format.
     */
    /**
     * - Parameter encoder: The encoder to write data to.
    */
public func encode any Encoder throws

    /**
     * Creates a new instance by decoding from the given decoder.
     */
    /**
     * This initializer throws an error if reading from the decoder fails, or
     * if the data read is corrupted or otherwise invalid.
     */
    /**
     * - Parameter decoder: The decoder to read data from.
    */
public init any Decoder throws
```

```
    /**
     * A compatibility error.
    */
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public enum CompatibilityError LocalizedError Equatable
Sendable
```

```
    /**
     * An error that indicates that the revision is not supported.
    */
case unsupportedRevision Int
```

```
    /// A localized message describing what error occurred.
    public var errorDescription String get

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func CompatibilityError
    CompatibilityError Bool

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension CompatibilityError CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription String get

    /// A transformer that composes two tabular transformers by applying them one
    after the other.
    ///
    /// The result of this transformer is equivalent to invoking `outer(inner(x))`
    on an input `x`,
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct ComposedTabularTransformer Inner Outer
TabularTransformer where Inner TabularTransformer Outer
TabularTransformer

    /// The inner transformer.
    public var inner Inner

    /// The outer transformer.
    public var outer Outer

    /// Creates a composed tabular transformer from two tabular transformers.
    /// - Parameters:
    ///   - inner: The inner transformer.
    ///   - outer: The outer transformer.
    public init _ Inner _ Outer

    /// Performs the composed transformation on a single input.
    ///
    /// - Parameters:
    ///   - input: The transformer input.
    ///   - eventHandler: An event handler.
```

```

    /// - Returns: An output produced by applying the transformer to the
    input.
    @inlinable public func applied           DataFrame
        EventHandler    nil  async throws   DataFrame

    /// The input type.
    @available iOS 16.0  tvOS 16.0  watchOS 11.0  macOS 13.0
    public typealias Input    DataFrame

    /// The output type.
    @available iOS 16.0  tvOS 16.0  watchOS 11.0  macOS 13.0
    public typealias Output   DataFrame

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension ComposedTabularTransformer
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`-
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription  String   get

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension ComposedTabularTransformer  Sendable where Inner
Sendable  Outer  Sendable

```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTabularTransformer Equatable where Inner
Equatable Outer Equatable

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
ComposedTabularTransformer Inner Outer
ComposedTabularTransformer Inner Outer Bool
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTabularTransformer Encodable where Inner
Encodable Outer Encodable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTabularTransformer Decodable where Inner
Decodable Outer Decodable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
    /// A temporal transformer that composes two temporal transformers by applying
```

them one after the other.

```
///  
/// The inner transformer's output must match the outer transformer input. The  
result of this transformer is equivalent  
/// to invoking `outer(inner(x))` on an input `x`,  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct ComposedTemporalTransformer Inner Outer  
TemporalTransformer where Inner TemporalTransformer Outer  
TemporalTransformer Inner Output Outer Input  
  
/// The input type.  
public typealias Input Inner Input  
  
/// The intermediate type.  
public typealias Intermediate Inner Output  
  
/// The output type.  
public typealias Output Outer Output  
  
/// The output sequence type.  
public typealias OutputSequence Outer OutputSequence  
  
/// The inner transformer.  
public var inner Inner  
  
/// The outer transformer.  
public var outer Outer  
  
/// Creates a transformer composition from two temporal transformers.  
/// - Parameters:  
///   - inner: The inner transformer.  
///   - outer: The outer transformer.  
public init _ Inner _ Outer  
  
/// Performs the composed transformation on an input sequence.  
///  
/// - Parameters:  
///   - input: The input temporal sequence.  
///   - eventHandler: An event handler.  
/// - Returns: An async sequence produced by applying the  
transformation to the input.  
@inlinable public func applied S S  
EventHandler nil async throws  
ComposedTemporalTransformer Inner Outer OutputSequence where  
S TemporalSequence Inner Input S Feature  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension ComposedTemporalTransformer  
CustomDebugStringConvertible
```

```

/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer. This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
///     struct Point: CustomDebugStringConvertible {
///         let x: Int, y: Int
///
///         var debugDescription: String {
///             return "(\(x), \(y))"
///         }
///     }
///
///     let p = Point(x: 21, y: 30)
///     let s = String(reflecting: p)
///     print(s)
///     // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTemporalTransformer Sendable where Inner
Sendable Outer Sendable

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTemporalTransformer Equatable where
Inner Equatable Outer Equatable

```

```

/// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:
///   - `lhs`: A value to compare.
///   - `rhs`: Another value to compare.
public static func
ComposedTemporalTransformer Inner Outer
ComposedTemporalTransformer Inner Outer Bool

```

```


@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTemporalTransformer : Encodable where
    Inner : Encodable, Outer : Encodable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
    public func encode(to: any Encoder) throws

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTemporalTransformer : Decodable where
    Inner : Decodable, Outer : Decodable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
    public init(from: any Decoder) throws

/// A transformer that composes two transformers by applying them one after the
other.
///
/// The inner transformer's output must match the outer transformer input. The
result of this transformer is equivalent
/// to invoking `outer(inner(x))` on an input `x`,
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct ComposedTransformer<Inner, Outer, Transformer>
where Inner : Transformer, Outer : Transformer, Inner.Output == Outer.Input

    /// The input type.
    public typealias Input = Inner.Input

    /// The intermediate type.
    public typealias Intermediate = Inner.Output

    /// The output type.
    public typealias Output = Outer.Output

    /// The inner transformer.


```

```

public var inner Inner
    /// The outer transformer.
public var outer Outer
    /// Creates a transformer composition from two transformers.
    /// - Parameters:
    ///   - inner: The inner transformer.
    ///   - outer: The outer transformer.
public init _ Inner _ Outer
    /// Performs the composed transformation on a single input.
    ///
    /// - Parameters:
    ///   - input: The transformer input.
    ///   - eventHandler: An event handler.
    /// - Returns: An output produced by applying the transformer to the

input.
@inlinable public func applied
ComposedTransformer Inner Outer Input
EventHandler nil async throws
ComposedTransformer Inner Outer Output

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ComposedTransformer CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    /// struct Point: CustomDebugStringConvertible {
    ///     let x: Int, y: Int
    ///
    ///     var debugDescription: String {
    ///         return "(\(x), \(y))"
    ///     }
    /// }
    ///
    /// let p = Point(x: 21, y: 30)
    /// let s = String(reflecting: p)
    /// print(s)
    /// // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the

```

```
/// `Point` type's `debugDescription` property.
public var debugDescription String get

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTransformer<Sendable> where Inner: Sendable, Outer: Sendable

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTransformer<Equatable> where Inner: Equatable, Outer: Equatable

/// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,  

/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:  

///   - lhs: A value to compare.  

///   - rhs: Another value to compare.
public static func equals(ComposedTransformer<Inner> Outer, ComposedTransformer<Inner> Outer) Bool

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTransformer<Encodable> where Inner: Encodable, Outer: Encodable

/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty  

/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given  

/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode(any Encoder) throws

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ComposedTransformer<Decodable> where Inner: Decodable, Outer: Decodable

/// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or  

/// if the data read is corrupted or otherwise invalid.
```

```

    /**
     * - Parameter decoder: The decoder to read data from.
    public init any Decoder throws

    /// Errors thrown when concatenating numeric values.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public enum ConcatenationError : LocalizedError, Equatable, Sendable

    /// A column contains arrays or shaped arrays with non-uniform shapes.
    case nonUniformShapes String

    /// Shaped arrays across columns have mismatched shapes and can't be concatenated.
    case mismatchedShapes

    /// A localized message describing what error occurred.
    public var errorDescription : String get

    /// Returns a Boolean value indicating whether two values are equal.
    /**
     * Equality is the inverse of inequality. For any values `a` and `b`,
     * `a == b` implies that `a != b` is `false`.
    /**
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func == (ConcatenationError ConcatenationError) Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ConcatenationError : CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription : String get

    /// Annotation parameters for the dataframe containing temporal annotations.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct DataFrameTemporalAnnotationParameters : Annotation where
    Annotation : Equatable, Annotation : Sendable

    /// The file path type to be used.
    public enum FilePathType : Equatable, CustomStringConvertible, Sendable

```

```

    /// The file path is relative to the `baseURL`.
case relative URL

    /// The file path is absolute.
case absolute

    /// A textual representation of this instance.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the
`String(describing:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `description` property for types that conform to
    /// `CustomStringConvertible`:
    ///
    ///     struct Point: CustomStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var description: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(describing: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `description` property.
public var description String get

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
DataFrameTemporalAnnotationParameters Annotation FilePathType

DataFrameTemporalAnnotationParameters Annotation FilePathType
Bool

    /// The file path type in the annotation file. The default value is
    `.absolute`.

```

```
public var filePathType  
DataFrameTemporalAnnotationParameters Annotation FilePathType  
  
    /// The column id that contains the file path. The default value is "filePath"  
    with String type.  
    public var filePathColumnID ColumnID String  
  
    /// The column id that contains the annotation. The default value is  
    "annotation" with `Annotation` type.  
    public var annotationColumnID ColumnID Annotation  
  
    /// The column id that contains the start time. The default value is `nil`.  
    public var startTimeColumnID ColumnID Double  
  
    /// The column id that contains the end time. The default value is `nil`.  
    public var endTimeColumnID ColumnID Double  
  
    /// Creates a DataFrameTemporalAnnotationParameters by using default  
    options.  
    public init
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension DataFrameTemporalAnnotationParameters Sendable  
  
    /// Dataset processing errors.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
    public enum DatasetError : LocalizedError, Equatable, Sendable
```

```
        /// An error that indicates that a resource is missing.  
        case missingResource URL  
  
        /// An error that indicates that a resource doesn't have the expected data  
        format.  
        case incompatibleDataFormat URL : String  
  
        /// An error that indicates that a resource is unreadable.  
        case unreadableResource URL  
  
        /// An error that indicates that a resource has incorrect name format.  
        case incorrectName URL : String  
  
        /// A localized message describing what error occurred.  
        public var errorDescription : String get  
  
        /// Returns a Boolean value indicating whether two values are equal.  
        ///  
        /// Equality is the inverse of inequality. For any values `a` and `b`,
```

```

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func DatasetError DatasetError
Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension DatasetError CustomDebugStringConvertible

    /// A text representation of the error.
public var debugDescription String get

    /// A time and date feature extractor.
    ///
    /// This transformer takes a
<doc://com.apple.documentation/documentation/foundation/date> and extracts
    /// floating-point feature values according to the features parameter. Every feature
value is roughly between -0.5 and
    /// 0.5. All date calculations are based on a
<doc://com.apple.documentation/documentation/foundation/calendar>,
    /// which defaults to
<doc://com.apple.documentation/documentation/foundation/calendar/2293438-
current>.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct DateFeatureExtractor Scalar Transformer
Sendable where Scalar BinaryFloatingPoint

    /// The calendar.
public let calendar Calendar

    /// The date and time features.
public let features DateFeatures

    /// Creates a date feature extractor.
    ///
    /// - Parameters:
    ///   - calendar: The calendar to use, defaults to
`Calendar.current`.
    ///   - features: The date and time features.
public init DateFeatures Calendar

    /// Extracts features of a particular date.
    ///

```

```

    /// - Parameters:
    ///   - date: The date.
    ///   - eventHandler: An event handler.
    /// - Returns: An array of feature values.
    public func applied Date
EventHandler nil Scalar

    /// The input type.
@available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0 macOS 15.0
public typealias Input Date

    /// The output type.
@available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0 macOS 15.0
public typealias Output Scalar

@available macOS 15.0 iOS 18.0 tvOS 18.0 watchOS 11.0
extension DateFeatureExtractor Codable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
    public init any Decoder throws

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
    public func encode any Encoder throws

    /// A set of date and time features.
    ///
    /// The choice of features for a particular task depends on the relevance of
    /// different date and time components. For
    /// example a dataset of weather data may require hour and day-of-year features,
    /// while a dataset of workout metrics may
    /// require second, hour, and weekday features.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0

```

```
watchOS 11.0
public struct DateFeatures : Hashable, OptionSet, RawRepresentable, Codable, Sendable

    /// The corresponding value of the raw type.
    ///
    /// A new instance initialized with `rawValue` will be equivalent to this
    /// instance. For example:
    ///
    ///     enum PaperSize: String {
    ///         case A4, A5, Letter, Legal
    ///     }
    ///
    ///     let selectedSize = PaperSize.Letter
    ///     print(selectedSize.rawValue)
    ///     // Prints "Letter"
    ///
    ///     print(selectedSize == PaperSize(rawValue:
    /// selectedSize.rawValue)!)
    ///     // Prints "true"
    public var rawValue: Int

    /// Creates a feature from a raw value.
    public init(rawValue: Int)

    /// A feature representing the second within the minute.
    public static let second: DateFeatures

    /// A feature representing the minute within the hour.
    public static let minute: DateFeatures

    /// A feature representing the hour within the day.
    public static let hour: DateFeatures

    /// A feature representing the weekday.
    public static let weekday: DateFeatures

    /// A feature representing the day within the month.
    public static let day: DateFeatures

    /// A feature representing the day within the year.
    public static let dayOfYear: DateFeatures

    /// A feature representing the week within the month.
    public static let weekOfMonth: DateFeatures

    /// A feature representing the week within the year.
    public static let weekOfYear: DateFeatures
```

```
    /// A feature representing the month within the year.
    public static let month DateFeatures

    /// The type of the elements of an array literal.
    @available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0  macOS 15.0
    public typealias ArrayLiteralElement DateFeatures

    /// The element type of the option set.
    ///
    /// To inherit all the default implementations from the `OptionSet` protocol,
    /// the `Element` type must be `Self`, the default.
    @available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0  macOS 15.0
    public typealias Element DateFeatures

    /// The raw type that can be used to represent all values of the conforming
    /// type.
    ///
    /// Every distinct value of the conforming type has a corresponding unique
    /// value of the `RawValue` type, but there may be values of the
    `RawValue`
    /// type that don't have a corresponding value of the conforming type.
    @available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0  macOS 15.0
    public typealias RawValue Int

    /// An item in a detection result.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public struct DetectedObject Label Equatable where Label : Comparable, Label : Hashable

        /// The bounding box of the detected object.
        public var boundingBox CGRect

        /// The detected object label.
        public var label Label

        /// The detection confidence. The value will always be between 0.0 and 1.0.
        public var confidence Float

        /// Creates a detected object with bounding box, object label and confidence.
        ///
        /// - Parameters:
        ///   - boundingBox: The bounding box of the detected object.
        ///   - label: The label of the detected object.
        ///   - probability: The detection confidence. The value will always be
        between 0.0 and 1.0
```

```

public init           CGRect      Label
Float

    /// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:
///   - lhs: A value to compare.
///   - rhs: Another value to compare.
public static func     DetectedObject Label
DetectedObject Label    Bool

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension DetectedObject Encodable where Label Encodable

    /// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode     any Encoder throws

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension DetectedObject Decodable where Label Decodable

    /// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.
///
/// - Parameter decoder: The decoder to read data from.
public init           any Decoder throws

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension DetectedObject Sendable where Label Sendable

    /// A temporal transformer that down samples the input stream.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct Downampler Input TemporalTransformer

```

```
Codable Sendable where Input Sendable

    /// The input type.
    public typealias Input Input

    /// The output type.
    public typealias Output Input

    /// The down sample factor to the input stream.
    public let factor Int

    /// Creates a down sample temporal transformer.
    /// - Parameters:
    ///   - factor: The down sample factor to the input stream.
    ///   - Precondition: `factor` must be greater than 0.
    public init Int

    /// Down samples the input sequence
    ///
    /// - Parameters:
    ///   - input: An async sequence of inputs.
    ///   - eventHandler: An event handler.
    /// - Returns: An async sequence of down sampled outputs.
    public func applied S S
EventHandler throws Downsampler Input DownStreamSequence
where Input S Feature S TemporalSequence

    /// The output async sequence type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias OutputSequence
Downsample Input DownStreamSequence

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
    public func encode any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
```

```
    /// - Parameter decoder: The decoder to read data from.
public init          any Decoder throws

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension Downampler

    /// An async sequence of down stream elements.
public struct DownStreamSequence  TemporalSequence

        /// The feature type.
public typealias Feature  Input

        /// The type of asynchronous iterator that produces elements of this
        /// asynchronous sequence.
public typealias AsyncIterator
Downampler Input  DownStreamSequence Iterator

        /// The count of elements.
public var count  Int  get

        /// Creates the asynchronous iterator that produces elements of this
        /// asynchronous sequence.
        ///
        /// - Returns: An instance of the `AsyncIterator` type used to
produce
        /// elements of the asynchronous sequence.
public func makeAsyncIterator
Downampler Input  DownStreamSequence Iterator

        /// The type of element produced by this asynchronous sequence.
@available iOS 16.0  tvOS 16.0  watchOS 11.0  macOS
13.0
        public typealias Element
TemporalFeature Downampler Input  DownStreamSequence Feature

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension Downampler DownStreamSequence

    /// An async iterator of down stream sequence.
public struct Iterator  AsyncIteratorProtocol

        /// Asynchronously advances to the next element and returns it, or
ends the
        /// sequence if there is no next element.
        ///
        /// - Returns: The next element, if it exists, or `nil` to signal the
end of
```

```

    /// the sequence.
    public mutating func next async throws
TemporalFeature Downampler Input Output

    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
    public typealias Element
TemporalFeature Downampler Input Output


/// An estimator that creates a transformer by fitting to a data set.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol Estimator

    /// The transformer type created by this estimator.
associatedtype Transformer Transformer

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    func fitted S S EventHandler
async throws Self Transformer where S Sequence S Element
Self Transformer Input

    /// Encodes a fitted transformer.
    func encode _ Self Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    func decode inout any EstimatorDecoder
throws Self Transformer


extension Estimator

    /// Exposes this estimator as a supervised estimator.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func
adaptedAsSupervised Annotation
Annotation self
EstimatorToSupervisedAdaptor Self Annotation where
Annotation Equatable

```

```
extension Estimator
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
    @inlinable public func fitted S S async throws  
    Self Transformer where S Sequence S Element  
    Self Transformer Input
```

```
extension Estimator
```

```
    /// Composes this estimator with a supervised temporal estimator.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available
```

```
    public func appending Other _ Other some  
SupervisedTemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer Other Transformer  
Other Annotation where Other SupervisedTemporalEstimator  
Self Transformer Output Other Transformer Input
```

```
extension Estimator
```

```
    /// Composes this estimator with a temporal transformer.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available
```

```
    public func appending Other _ Other some  
TemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer Other where Other  
TemporalTransformer Other Input Self Transformer Output
```

```
    /// Composes this estimator with a temporal estimator.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available
```

```
    public func appending Other _ Other some  
TemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer Other Transformer where  
Other TemporalEstimator Self Transformer Output
```

Other Transformer Input

extension Estimator

```
/// Composes this estimator with a transformer.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other -> Other some  
Estimator ComposedTransformer Self Transformer Other where  
Other Transformer Other Input Self Transformer Output
```

```
/// Composes this estimator with another estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other -> Other some  
Estimator ComposedTransformer Self Transformer  
Other Transformer where Other Estimator  
Self Transformer Output Other Transformer Input
```

extension Estimator

```
/// Exposes this estimator as a temporal estimator.
```

```
@available iOS 13.0 tvOS 15.0
```

```
@available iOS 16.0 tvOS 18.0
```

```
@available iOS 16.0 tvOS 18.0
```

```
@available iOS 1.0 tvOS 2.0
```

```
@available
```

```
@inlinable public func adaptedAsTemporal
```

```
EstimatorToTemporalAdaptor Self
```

extension Estimator

```
/// Composes this estimator with a supervised estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other -> Other some  
SupervisedEstimator ComposedTransformer Self Transformer  
Other Transformer Other Annotation where Other  
SupervisedEstimator Self Transformer Output  
Other Transformer Input
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Estimator where Self Transformer Encodable
```

```
    /// Encodes a fitted encodable transformer.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - encoder: An estimator encoder.
    public func encode _ Self Transformer
        inout any EstimatorEncoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Estimator where Self Transformer Decodable
```

```
    /// Decodes a previously fitted decodable transformer.
    ///
    /// - Parameter decoder: An estimator decoder.
    /// - Returns: The decoded transformer.
    public func decode inout any
        EstimatorDecoder throws Self Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Estimator
```

```
    /// Writes the encoded transformer to a file.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - url: A file URL.
    ///   - overwrite: A Boolean value indicating whether to overwrite
    ///     existing files.
    public func write _ Self Transformer
        URL Bool true throws

    /// Reads the encoded transformer from a file.
    ///
    /// - Parameter url: A file URL.
    /// - Returns: The decoded transformer.
    public func read URL throws Self Transformer
```

```
/// A type that can decode values from a model representation.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol EstimatorDecoder
```

```
    /// Decodes a value.
    mutating func decode T _ T throws T where
```

```
T Decodable
```

```
    /// Decodes an optimizer value.  
    ///  
    /// Decoding an optimizer lets you resume fitting.  
    mutating func decodeOptimizer T _           T      throws  
T where T Decodable
```

```
    /// A type that can encode values into a model representation.  
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0  
public protocol EstimatorEncoder
```

```
    /// Encodes a value.  
    mutating func encode T _           T      throws where T  
Encodable
```

```
    /// Encodes an estimator optimizer.  
    ///  
    /// Optimizers are used when fitting an estimator and usually contain state  
information such as momentum.  
    /// This method encodes the optimizer state separately from model  
parameters.  
    mutating func encodeOptimizer T _           T      throws where  
T Encodable
```

```
    /// An estimator encoding error.  
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0  
public enum EstimatorEncodingError : LocalizedError  
Equatable Sendable
```

```
    /// An error that indicates that an estimator cannot perform encoding from its  
current state.
```

```
    case invalidState : String
```

```
    /// A localized message describing what error occurred.
```

```
    public var errorDescription : String { get }
```

```
    /// Returns a Boolean value indicating whether two values are equal.
```

```
    ///
```

```
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.
```

```
    ///
```

```
    /// - Parameters:
```

```
    ///     - lhs: A value to compare.
```

```
    ///     - rhs: Another value to compare.
```

```
    public static func EstimatorEncodingError
```

```
EstimatorEncodingError Bool
```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension EstimatorEncodingError
CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription String get

    /// An adaptor that exposes an estimator as a supervised estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct EstimatorToSupervisedAdaptor<Estimator>
    : Annotation<SupervisedEstimator<Estimator>>, where Estimator : Estimator
    : Annotation<Equatable>

    /// The transformer type created by this estimator.
    public typealias Transformer = Estimator Transformer

    /// The wrapped estimator.
    public let estimator: Estimator

    /// Creates an estimator adaptor.
    public init(_ estimator: Estimator)

    /// Fits a transformer to a sequence of examples, ignoring the annotations
    and the validation.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The pre-defined transformer.
    public func fitted<Input: Sequence<InputElement>, InputElement: AnnotatedFeature<Estimator, Transformer, Input, Annotation>>()
        async throws
            EstimatorToSupervisedAdaptor<Estimator>
            Annotation<Transformer<where Input: Sequence<InputElement>, InputElement: AnnotatedFeature<Estimator, Transformer, Input, Annotation>>

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
    fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted<Input: Sequence<InputValidation>, InputValidation: Validation<Input>, InputElement: AnnotatedFeature<EstimatorToSupervisedAdaptor<Estimator>, Validation: Sequence<Validation>, ValidationElement: InputElement>>()
        nil async throws
            EstimatorToSupervisedAdaptor<Estimator>
            Annotation<Transformer<where Input: Sequence<Validation>, Validation: Sequence<ValidationElement>>

```

```

AnnotatedFeature Estimator Transformer Input Annotation
Validation Element
AnnotatedFeature Estimator Transformer Input Annotation

    /// Does nothing since this estimator uses a pre-defined transformer.
    public func encode _  

EstimatorToSupervisedAdaptor Estimator
Annotation Transformer inout any
EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    public func decode inout any  

EstimatorDecoder throws
EstimatorToSupervisedAdaptor Estimator
Annotation Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension EstimatorToSupervisedAdaptor Sendable where
Estimator Sendable Annotation Sendable

    /// A temporal estimator wrapping an estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public struct EstimatorToTemporalAdaptor Base
TemporalEstimator where Base Estimator

    /// The transformer type created by this estimator.
    public typealias Transformer
TransformerToTemporalAdaptor Base Transformer

    /// The input type.
    public typealias Input Base Transformer Input

    /// The output type.
    public typealias Output Base Transformer Output

    /// Creates a temporal estimator from an estimator.
    ///
    /// The resulting estimator collects all elements of the input sequence before
    calling fit on the underlying
    /// estimator. The transformer returned from fit is also converted to a
temporal transformer.
    public init _ Base

    /// Fits a transformer to a sequence of examples.

```

```

    /**
     * - Parameters:
     *   - input: A sequence of examples.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     */
    @inlinable public func fitted<InputSequence: Sequence<InputSequence<Element>, EventHandler: Base<Transformer<Input<TemporalSequence<Base<Transformer<Input<Element>, Feature>>>> nil async throws
        EstimatorToTemporalAdaptor<Base<Transformer<Input<TemporalSequence<Base<Transformer<Input<Element>, Feature>>>>
        InputSequence<Sequence<InputSequence<Element>>
        TemporalSequence<Base<Transformer<Input<Element>, Feature>>
        InputSequence<Element>

    /**
     * Encodes a fitted transformer.
     */
    @inlinable public func encode<EstimatorToTemporalAdaptor<Base<Transformer<inout any<EstimatorEncoder>>> throws
        /// Decodes the transformer.
        @inlinable public func decode<EstimatorDecoder> inout any<EstimatorToTemporalAdaptor<Base<Transformer>>>
        EstimatorDecoder<throws>

    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    extension EstimatorToTemporalAdaptor<Sendable where Base: Sendable>

```

```

    /**
     * Maintains the status of the pipeline.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    public struct Event<Sendable>
    {
        /**
         * A description of the event's origin.
         */
        public var origin: String

        /**
         * The number of items processed so far.
         */
        public var itemCount: Int

        /**
         * The total number of items being processed.
         */
        public var totalItemCount: Int

        /**
         * A dictionary of custom metrics values.
         */
        public var metrics: MetricsKey<any Sendable>

        /**
         * Creates an event.
         */
    }

```

```

////
//// - Parameters:
//// - origin: A description of the event's origin.
//// - itemCount: The number of items processed so far.
//// - totalItemCount: The total number of items being processed.
//// - metrics: A dictionary of custom metrics values.
public init String Int
          Int nil MetricsKey any
Sendable

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Event CustomDebugStringConvertible

/// A textual representation of this instance, suitable for debugging.
////
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` 
/// initializer. This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
////
/// struct Point: CustomDebugStringConvertible {
///     let x: Int, y: Int
////
///     var debugDescription: String {
///         return "(\(x), \(y))"
///     }
///}
////
/// let p = Point(x: 21, y: 30)
/// let s = String(reflecting: p)
/// print(s)
/// // Prints "(21, 30)"
////
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

/// A closure to handle processing events.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public typealias EventHandler @Sendable Event Void

/// A classifier that uses a fully connected network.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct FullyConnectedNetworkClassifier Scalar Label
SupervisedEstimator where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint Label Comparable Label Decodable

```

```
Label Encodable Label Hashable

    /// The transformer type created by this estimator.
    public typealias Transformer
FullyConnectedNetworkClassifierModel Scalar Label

    /// The annotation type.
    public typealias Annotation Label

    /// The fully-connected-network configuration.
    public var configuration
FullyConnectedNetworkConfiguration

    /// The set of possible labels.
    public var labels Set Label

    /// Creates a fully connected network classifier.
    ///
    /// - Parameters:
    ///   - labels: The labels used to train the classifier.
    ///   - configuration: The configuration.
    public init Set Label
FullyConnectedNetworkConfiguration

    /// Fits a fully connected network classifier model to a sequence of
examples.
    ///
    /// The training process partitions the input into random batches according to
the batch size configuration
    /// parameter. Training stops when the maximum number of iterations is
reached.
    ///
    /// - Note: This method does not do early-stopping, using a high value for
`maximumIterations` may lead to
    /// over-fitting. Consider providing a validation set.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the classifier.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted fully connected network classifier model.
    public func fitted Input Input
EventHandler nil async throws
FullyConnectedNetworkClassifierModel Scalar Label where
Input Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Label

    /// Fits a fully connected network classifier model to a sequence of
examples.
    ///
    /// The training process partitions the input into random batches according to
```

```
the batch size configuration
    /// parameter. Training stops when the validation loss stops improving or
when the maximum number of iterations
    /// is reached.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the classifier.
    ///   - validation: A sequence of examples used for validating the
fitted classifier.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted fully connected network classifier model.
public func fitted Input Validation Input
                    Validation EventHandler
nil async throws
FullyConnectedNetworkClassifierModel Scalar Label where
Input Sequence Validation Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Label
Validation Element AnnotatedFeature MLShapedArray Scalar
Label
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension FullyConnectedNetworkClassifier

    /// Decodes the estimator.
    /// - Parameter decoder: A decoder for the estimator.
    /// - Returns: A fully connected network classifier model.
public func decode inout any
EstimatorDecoder throws
FullyConnectedNetworkClassifierModel Scalar Label

    /// Encodes a fitted transformer with an optimizer.
    ///
    /// - Parameters:
    ///   - transformer: A fully connected network classifier model.
    ///   - encoder: An encoder for the estimator.
public func encodeWithOptimizer _
FullyConnectedNetworkClassifier Scalar Label Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer with an optimizer.
    ///
    /// - Parameter decoder: A decoder for the estimator.
    /// - Returns: A fully connected network classifier model.
public func decodeWithOptimizer inout any
EstimatorDecoder throws
FullyConnectedNetworkClassifier Scalar Label Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
extension FullyConnectedNetworkClassifier  
UpdatableSupervisedEstimator

    /// Creates a default-initialized fully connected network classifier model  
suitable for incremental fitting.
    public func makeTransformer  
FullyConnectedNetworkClassifierModel Scalar Label

    /// Updates a fully connected network classifier model with a new sequence  
of examples.
    ///
    /// - Parameters:
    /// - transformer: A fully connected network classifier model to  
update.
    /// - input: A sequence of examples.
    /// - eventHandler: An event handler.
    public func update InputSequence _ inout  
FullyConnectedNetworkClassifierModel Scalar Label
        InputSequence EventHandler nil async
throws where InputSequence Sequence InputSequence Element
        AnnotatedFeature MLShapedArray Scalar Label
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension FullyConnectedNetworkClassifier Sendable where
Scalar Sendable Label Sendable
```

```
/// A classifier model that uses a fully connected network.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct FullyConnectedNetworkClassifierModel Scalar
Label Classifier where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint Label Comparable Label Decodable
Label Encodable Label Hashable

    /// Performs a classification on a shaped array.
    ///
    /// - Parameters:
    /// - input: The classifier input.
    /// - eventHandler: An event handler.
    /// - Returns: A classification distribution.
    public func applied MLShapedArray Scalar
        EventHandler nil async throws
ClassificationDistribution Label

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Input MLShapedArray Scalar
```

```
    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Output
    ClassificationDistribution Label

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    extension FullyConnectedNetworkClassifierModel Codable

        /// Encodes this value into the given encoder.
        ///
        /// If the value fails to encode anything, `encoder` will encode an empty
        /// keyed container in its place.
        ///
        /// This function throws an error if any values are invalid for the given
        /// encoder's format.
        ///
        /// - Parameter encoder: The encoder to write data to.
    public func encode           any Encoder throws

        /// Creates a new instance by decoding from the given decoder.
        ///
        /// This initializer throws an error if reading from the decoder fails, or
        /// if the data read is corrupted or otherwise invalid.
        ///
        /// - Parameter decoder: The decoder to read data from.
    public init           any Decoder throws

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    extension FullyConnectedNetworkClassifierModel
    CustomDebugStringConvertible

        /// A textual representation of this instance, suitable for debugging.
        ///
        /// Calling this property directly is discouraged. Instead, convert an
        /// instance of any type to a string by using the `String(reflecting:)`
        /// initializer. This initializer works with any type, and uses the custom
        /// `debugDescription` property for types that conform to
        /// `CustomDebugStringConvertible`.
        ///
        ///     struct Point: CustomDebugStringConvertible {
        ///         let x: Int, y: Int
        ///
        ///         var debugDescription: String {
        ///             return "(\(x), \(y))"
        ///         }
        ///     }

```

```

    /**
     *      let p = Point(x: 21, y: 30)
     *      let s = String(describing: p)
     *      print(s)
     *      // Prints "(21, 30)"
     *
     *      The conversion of `p` to a string in the assignment to `s` uses the
     *      `Point` type's `debugDescription` property.
  public var debugDescription String get

    /// A fully connected network configuration.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct FullyConnectedNetworkConfiguration : Hashable
Codable Sendable

    /// The number of neurons in each hidden layer.
    /**
     *      Defaults to a single hidden layer with 100 neurons.
  public var hiddenUnitCounts Int

    /// The learning rate.
    /**
     *      The learning rate controls how much the model changes when presented
     *      with new data. A high learning rate may
     *          overshoot when close to a solution, while a low learning rate may take too
     *      long to train a good model.
     *
     *      Defaults to 0.001.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public var learningRate Float

    /// The maximum number of iterations.
    /**
     *      More iterations will produce better models as long as there is no over-
     *      fitting. Over-fitting happens when the
     *          dropout probability is too low or there is not enough training data.
     *
     *      - Note: This parameter is only used by the `fitted` method. When
     *      using the `update` method it's up to you
     *          to decide when to stop.
  public var maximumIterations Int

    /// The dropout probability.
    /**
     *      Dropout layers are placed after fully-connected layers to help prevent
     *      over-fitting.
     *
     *      Defaults to 0.2.
@available macOS 14.0 iOS 17.0 tvOS 17.0

```

```
public var dropoutProbability  Float
    /**
     * The number of examples to use per mini-batch.
     */
    /**
     * A larger batch size will speed up computation when training, as long as
     * the batch fits in memory.
     */
    /**
     * - Note: This parameter is only used by the `fitted` method. When
     * using the `update` method it's up to you
     *      to do batching.
     */
public var batchSize  Int

    /**
     * The early-stopping tolerance.
     */
    /**
     * The tolerance is used by the `fitted` method to decide when progress
     * is no longer being made, in which case the
     *      training process will stop before the specified maximum number of
     * iterations (known as early stopping).
     */
    /**
     * Significant progress happens when the validation loss decreases by at
     * least the tolerance.
     */
    /**
     * Defaults to 0.01.
     */
    /**
     * - Note: Early stopping only happens when using the `fitted` method
     * with validation data.
     */
public var earlyStoppingTolerance  Double

    /**
     * The number of iterations to use when evaluating whether to stop early.
     */
    /**
     * The `fitted` method will stop if no significant progress is made for this
     * many iterations. Significant
     *      progress happens when the validation loss decreases by at least
     * `earlyStoppingTolerance`.
     */
    /**
     * Defaults to 10.
     */
    /**
     * - Note: Early stopping only happens when using the `fitted` method
     * with validation data.
     */
@available macOS 14.0  iOS 17.0  tvOS 17.0
public var earlyStopIterationCount  Int

    /**
     * A seed to generate reproducible results from random operations such as
     * column and row subsampling.
     */
public var randomSeed  Int

    /**
     * Creates a default fully-connected-network configuration.
     */
public init

    /**
     * Hashes the essential components of this value by feeding them into the
     * given hasher.
     */
```

```
///  
/// Implement this method to conform to the `Hashable` protocol. The  
/// components used for hashing must be the same as the components  
compared  
/// in your type's `==` operator implementation. Call  
`hasher.combine(_:)`  
/// with each of these components.  
///  
/// - Important: In your implementation of `hash(into:)`,  
/// don't call `finalize()` on the `hasher` instance provided,  
/// or replace it with a different instance.  
/// Doing so may become a compile-time error in the future.  
///  
/// - Parameter hasher: The hasher to use when combining the  
components  
/// of this instance.  
public func hash inout Hasher
```

/// Returns a Boolean value indicating whether two values are equal.

///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.

/// - **Parameters:**

/// - lhs: A value to compare.
/// - rhs: Another value to compare.

public static func

FullyConnectedNetworkConfiguration FullyConnectedNetworkConfiguration Bool

/// Encodes this value into the given encoder.

///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.

///
/// This function throws an error if any values are invalid for the given
/// encoder's format.

/// - **Parameter** encoder: The encoder to write data to.
public func encode **any** Encoder **throws**

/// The hash value.

///
/// Hash values are not guaranteed to be equal across different executions of
/// your program. Do not save hash values to use during a future execution.

///
/// - **Important:** `hashValue` is deprecated as a `Hashable`
requirement. To
/// conform to `Hashable`, implement the `hash(into:)` requirement
instead.

```

    ///> The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

    ///> Creates a new instance by decoding from the given decoder.
    ///
    ///> This initializer throws an error if reading from the decoder fails, or
    ///> if the data read is corrupted or otherwise invalid.
    ///
    ///> - Parameter decoder: The decoder to read data from.
public init any Decoder throws

    ///> A classifier that uses a multi-label fully-connected network.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public struct
FullyConnectedNetworkMultiLabelClassifier Scalar Label
SupervisedEstimator where Scalar: MLShapedArrayScalar, Scalar
    : BinaryFloatingPoint, Scalar: Decodable, Scalar: Encodable
    Label: Comparable, Label: Decodable, Label: Encodable
    Label: Hashable

    ///> The transformer type created by this estimator.
public typealias Transformer
FullyConnectedNetworkMultiLabelClassifierModel Scalar Label

    ///> The annotation type.
public typealias Annotation Set Label

    ///> The fully-connected network configuration.
public var configuration
FullyConnectedNetworkConfiguration

    ///> The set of possible labels.
public var labels Set Label

    ///> The default fully-connected network configuration.
public static var defaultConfiguration
FullyConnectedNetworkConfiguration get

    ///> Creates a full-connected network multi-label classifier.
    ///
    ///> - Parameters:
    ///>   - labels: The labels used to train the multi-label classifier.
    ///>   - configuration: The configuration.
public init Set Label
FullyConnectedNetworkConfiguration

    ///> Fits a fully-connected network multi-label classifier model to a sequence
    of examples.

```

```

    /**
     * The training process partitions the input into random batches according to
     * the batch size configuration
     * parameter. Training stops when the maximum number of iterations is
     * reached.
     */
     /**
      * - Note: This method does not do early-stopping, using a high value for
      * `maximumIterations` may lead to
      * over-fitting. Consider providing a validation set.
      */
      /**
       * - Parameters:
       * - input: A sequence of examples used for fitting the classifier.
       * - eventHandler: An event handler.
       * - Returns: The fitted fully-connected network multi-label classifier
       * model.
       public func fitted Input EventHandler nil async throws
       FullyConnectedNetworkMultiLabelClassifierModel Scalar Label
       where Input Sequence Input Element
       AnnotatedFeature MLShapedArray Scalar Set Label

       /**
        * Fits a fully-connected network multi-label classifier model to a sequence
        * of examples.
        */
        /**
         * The training process partitions the input into random batches according to
         * the batch size configuration
         * parameter. Training stops when the validation loss stops improving or
         * when the maximum number of iterations
         * is reached.
         */
         /**
          * - Parameters:
          * - input: A sequence of examples used for fitting the classifier.
          * - validation: A sequence of examples used for validating the
          * fitted multi-label classifier.
          * - eventHandler: An event handler.
          * - Returns: The fitted fully-connected network multi-label classifier
          * model.
          public func fitted Input Validation EventHandler
          nil async throws
          FullyConnectedNetworkMultiLabelClassifierModel Scalar Label
          where Input Sequence Validation Sequence Input Element
          AnnotatedFeature MLShapedArray Scalar Set Label
          Validation Element AnnotatedFeature MLShapedArray Scalar
          Set Label

```

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifier

```

    /// Decodes the estimator.
    ///
    /// - Parameter decoder: A decoder for the estimator.
    /// - Returns: A fully-connected network multi-label classifier model.
    public func decode inout any
EstimatorDecoder throws
FullyConnectedNetworkMultiLabelClassifierModel Scalar Label

    /// Encodes a fitted transformer with an optimizer.
    ///
    /// - Parameters:
    ///   - transformer: A fully-connected network multi-label classifier
    model.
    ///   - encoder: An encoder for the estimator.
    public func encodeWithOptimizer -
FullyConnectedNetworkMultiLabelClassifier Scalar
Label Transformer inout any EstimatorEncoder
throws

    /// Decodes a previously fitted transformer with an optimizer.
    ///
    /// - Parameter decoder: A decoder for the estimator.
    /// - Returns: A fully-connected network multi-label classifier model.
    public func decodeWithOptimizer inout any
EstimatorDecoder throws
FullyConnectedNetworkMultiLabelClassifier Scalar
Label Transformer

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifier Sendable
where Scalar Sendable Label Sendable

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifier
UpdatableSupervisedEstimator

    /// Creates a default-initialized fully-connected network multi-label classifier
    model suitable for incremental fitting.
    public func makeTransformer
FullyConnectedNetworkMultiLabelClassifierModel Scalar Label

    /// Updates a fully-connected network multi-label classifier model with a new
    sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A fully-connected network multi-label classifier
    model to update.
    ///   - input: A sequence of examples.

```

```

    /// - eventHandler: An event handler.
    public func update InputSequence _ inout
FullyConnectedNetworkMultiLabelClassifierModel Scalar Label
    InputSequence EventHandler nil
async throws where InputSequence Sequence
InputSequence Element
AnnotatedFeature MLShapedArray Scalar Set Label

/// A multi-label classifier model that uses a fully-connected network.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public struct
FullyConnectedNetworkMultiLabelClassifierModel Scalar
Label Transformer where Scalar MLShapedArrayScalar
Scalar BinaryFloatingPoint Label Comparable Label
Decodable Label Encodable Label Hashable

    /// Updates the per-label precision-recall curve using the input data.
    ///
    /// Call this method before exporting to a Core ML Model and using Vision
`VNCoreMLRequest` to make predictions.
    ///
    /// - Parameter input: A collection of annotated examples.
    public mutating func updatePrecisionRecallCurves _
some Collection AnnotatedFeature MLShapedArray Scalar
Set Label async throws

    /// Performs a prediction and keeps label-confidence pairs that are greater
than or equal to the provided
    /// confidence thresholds.
    ///
    /// When the confidence threshold is `NaN`, the label-confidence pair is not
included in the
    /// result, regardless of the label's confidence.
    ///
    /// - Parameters:
    ///   - input: The classifier input.
    ///   - confidenceThresholds: A dictionary of label and confidence
threshold pairs.
    /// - Returns: A dictionary of labels and confidences.
    public func prediction
FullyConnectedNetworkMultiLabelClassifierModel Scalar
Label Input Label Scalar throws
ClassificationDistribution Label

    /// Performs a sequence of predictions and keeps label-confidence pairs that
are greater than or equal to the provided
    /// confidence thresholds.
    ///
    ///

```

```

    /// When the confidence threshold is `NaN`, the label-confidence pair is not
    included in the
    /// result, regardless of the label's confidence.
    ///
    /// - Parameters:
    ///   - input: A sequence of model inputs.
    ///   - confidenceThresholds: A dictionary of label and confidence
    threshold pairs.
    /// - Returns: An array of classification distributions.
public func prediction S S
    Label Scalar throws
ClassificationDistribution Label where S Sequence
S Element MLShapedArray Scalar

    /// Performs a classification on a shaped array.
    ///
    /// - Parameters:
    ///   - input: The classifier input.
    ///   - eventHandler: An event handler.
    /// - Returns: A classification distribution.
public func applied MLShapedArray Scalar
    EventHandler nil throws
ClassificationDistribution Label

    /// The input type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Input MLShapedArray Scalar

    /// The output type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Output
ClassificationDistribution Label

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifierModel
Codable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

    /// Creates a new instance by decoding from the given decoder.

```

```

////
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.
///
/// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifierModel
Sendable where Scalar Sendable Label Sendable

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifierModel
CustomDebugStringConvertible

/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer.
/// This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
///     struct Point: CustomDebugStringConvertible {
///         let x: Int, y: Int
///
///         var debugDescription: String {
///             return "(\(x), \(y))"
///         }
///     }
///
///     let p = Point(x: 21, y: 30)
///     let s = String(reflecting: p)
///     print(s)
///     // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension FullyConnectedNetworkMultiLabelClassifierModel

/// Computes evaluation metrics on annotated examples.
///
/// - Parameters:

```

```

    /// - input: A collection of annotated examples.
    /// - confidenceThresholds: A dictionary of label and confidence
threshold pairs.
    /// - Returns: Multi-label classifier metrics.
public func evaluation some
Collection AnnotatedFeature MLShapedArray Scalar
Set Label Label Float throws
MultiLabelClassificationMetrics Label

/// A regressor that uses a fully connected network.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct FullyConnectedNetworkRegressor Scalar
SupervisedEstimator where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint

/// The transformer type created by this estimator.
public typealias Transformer
FullyConnectedNetworkRegressorModel Scalar

/// The annotation type.
public typealias Annotation Float

/// The fully-connected-network configuration.
public var configuration
FullyConnectedNetworkConfiguration

/// Creates a fully connected network regressor.
///
/// - Parameter configuration: The configuration.
public init
FullyConnectedNetworkConfiguration

/// Fits a fully connected network regressor model to a sequence of
examples.
///
/// The training process partitions the input into random batches according to
the batch size configuration
/// parameter. Training stops when the maximum number of iterations is
reached.
///
/// - Note: This method does not do early-stopping, using a high value for
`maximumIterations` may lead to
/// over-fitting. Consider providing a validation set.
///
/// - Parameters:
///   - input: A sequence of examples used for fitting the transformer.
///   - eventHandler: An event handler.
/// - Returns: The fitted transformer.
public func fitted Input Input

```

```

EventHandler nil async throws
FullyConnectedNetworkRegressor Scalar Transformer where Input
Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Float

    /// Fits a fully connected network regressor model to a sequence of
examples.
    ///
    /// The training process partitions the input into random batches according to
the batch size configuration
    /// parameter. Training stops when the validation loss stops improving or
when the maximum number of iterations
    /// is reached.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the regressor.
    ///   - validation: A sequence of examples used for validating the
fitted regressor.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted fully connected network regressor model.
public func fitted Input Validation Input
Validation EventHandler

nil async throws
FullyConnectedNetworkRegressorModel Scalar where Input
Sequence Validation Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Float
Validation Element AnnotatedFeature MLShapedArray Scalar
Float

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension FullyConnectedNetworkRegressor

```

    /// Decodes the estimator.
    /// - Parameter decoder: A decoder for the estimator.
    /// - Returns: A fully connected network regressor model.
public func decode inout any
EstimatorDecoder throws
FullyConnectedNetworkRegressorModel Scalar

    /// Encodes a fitted transformer with an optimizer.
    ///
    /// - Parameters:
    ///   - transformer: A fully connected network regressor model.
    ///   - encoder: An encoder for the estimator.
public func encodeWithOptimizer -
FullyConnectedNetworkRegressor Scalar Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer with an optimizer.

```

```


    /**
     * - Parameter decoder: A decoder for the estimator.
     * - Returns: A fully connected network regressor model.
     */
    public func decodeWithOptimizer           inout any
EstimatorDecoder throws
FullyConnectedNetworkRegressor Scalar Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension FullyConnectedNetworkRegressor Sendable where
Scalar Sendable

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension FullyConnectedNetworkRegressor
UpdatableSupervisedEstimator

    /// Creates a default-initialized fully connected network regressor model
    /// suitable for incremental fitting.
    public func makeTransformer
FullyConnectedNetworkRegressorModel Scalar

    /// Updates a fully connected network regressor model with a new sequence
    /// of examples.
    /**
     * - Parameters:
     *   - transformer: A fully connected network regressor model to
     *     update.
     *   - input: A sequence of examples.
     *   - eventHandler: An event handler.
     */
    public func update InputSequence _           inout
FullyConnectedNetworkRegressorModel Scalar
InputSequence           EventHandler nil async throws
where InputSequence Sequence InputSequence Element
AnnotatedFeature MLShapedArray Scalar Float

    /// A regressor model that uses a fully connected network.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct FullyConnectedNetworkRegressorModel Scalar
Regressor where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint

    public typealias Target Float

    /// Performs regression on a shaped array.
    /**
     * - Parameters:
     *   - input: The regressor input.
     *   - eventHandler: An event handler.
     */


```

```
    /// - Returns: A regression.
    public func applied(MLShapedArray Scalar,
                        EventHandler nil) async throws
FullyConnectedNetworkRegressorModel Scalar Target

    /// The input type.
    @available(iOS 16.0, tvOS 16.0, watchOS 11.0, macOS 13.0)
public typealias Input = MLShapedArray

    /// The output type.
    @available(iOS 16.0, tvOS 16.0, watchOS 11.0, macOS 13.0)
public typealias Output
FullyConnectedNetworkRegressorModel Scalar Target

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension FullyConnectedNetworkRegressorModel : Codable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
    public func encode(any Encoder) throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
    public init(any Decoder) throws

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension FullyConnectedNetworkRegressorModel : CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
```

```

/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
/// struct Point: CustomDebugStringConvertible {
///     let x: Int, y: Int
///
///     var debugDescription: String {
///         return "(\(x), \(y))"
///     }
/// }
///
/// let p = Point(x: 21, y: 30)
/// let s = String(reflecting: p)
/// print(s)
/// // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription: String { get

```

```

/// A human body action repetition counting transformer that takes window of
/// human body poses and produces cumulative human body action repetition counts.
@available(macOS 13.0, iOS 16.0, tvOS 16.0)
public struct HumanBodyActionCounter : TemporalTransformer<Sendable>
{
    /// The input type.
    public typealias Input = Pose

    /// The output type.
    public typealias Output = Float

    /// The output async sequence type.
    public typealias OutputSequence = HumanBodyActionCounter.CumulativeSumSequence

    /// Creates a human body action counter.
    public init()

    /// Predicts cumulative human body action counts from a sequence of human
    /// body pose windows.
    ///
    /// - Parameters:
    ///   - input: An async sequence of human body pose windows.
    ///   - eventHandler: An event handler.
    /// - Returns: An async sequence of cumulative human body action
    /// counts.
    public func applied<S>(EventHandler<S> nil, async throws

```

```
HumanBodyActionCounter OutputSequence where S
TemporalSequence S Feature Pose

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionCounter

    /// Cumulative human body action count sequence.
    public struct CumulativeSumSequence : TemporalSequence

        /// The type of asynchronous iterator that produces elements of this
        /// asynchronous sequence.
        public typealias AsyncIterator
HumanBodyActionCounter CumulativeSumSequence Iterator

        /// The feature type.
        public typealias Feature = Float

        /// The estimated number of predictions.
        public var count : Int get

        /// Constructs an iterator.
        public func makeAsyncIterator
HumanBodyActionCounter CumulativeSumSequence Iterator

        /// The type of element produced by this asynchronous sequence.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Element
TemporalFeature HumanBodyActionCounter CumulativeSumSequence Feature

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionCounter CumulativeSumSequence

    /// An async iterator of cumulative count sequence.
    public struct Iterator : AsyncIteratorProtocol

        /// Advances to the next element and returns it, or nil if no next element
        exists.
        public mutating func next async throws
TemporalFeature Float

@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Element = TemporalFeature Feature
```

```
/// A human body action period predictor transformer that takes window of poses  
and produces a window of predictions.  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public struct HumanBodyActionPeriodPredictor : Transformer  
Sendable  
  
    /// The input type.  
public typealias Input = Pose  
  
    /// The output type.  
public typealias Output = Prediction  
HumanBodyActionPeriodPredictor  
  
    /// Creates a human body action period predictor transformer.  
public init  
  
    /// Predicts human body action periods from an array of poses.  
    ///  
    /// - Parameters:  
    /// - input: An async sequence of pose windows.  
    /// - eventHandler: An event handler.  
    /// - Returns: An async sequence of predictions.  
    public func applied(_ input: [Pose],  
                        eventHandler: EventHandler) nil async throws  
    HumanBodyActionPeriodPredictor Prediction  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
extension HumanBodyActionPeriodPredictor  
  
    /// A human body action period prediction.  
public struct Prediction  
  
        /// The duration of a human body action measured in frames.  
public var period: Float  
  
        /// A score that indicates whether this frame contributes to a periodic  
        /// human body action.  
public var periodicity: Float  
  
        /// Creates a human body action period prediction.  
        ///  
        /// - parameters:  
        /// - period: A period length of a repetitive action.  
        /// - periodicity: A value that indicates whether a frame  
        /// belongs to a periodic action.  
public init(period: Float, periodicity: Float)
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionPeriodPredictor Prediction
Encodable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionPeriodPredictor Prediction
Decodable
```

```
    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionPeriodPredictor Prediction
Equatable
```

```
    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b` ,
    /// `a == b` implies that `a != b` is `false` .
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
HumanBodyActionPeriodPredictor Prediction
HumanBodyActionPeriodPredictor Prediction Bool
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionPeriodPredictor Prediction Hashable
```

```

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
    /// in your type's `==` operator implementation. Call
    `hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,
    /// don't call `finalize()` on the `hasher` instance provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
    components
    /// of this instance.
public func hash           inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`  

requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyActionPeriodPredictor Prediction Sendable

```

```

    /// The human body pose image feature extractor.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct HumanBodyPoseExtractor Transformer Sendable

    /// Creates a human body pose extractor transformer
public init

    /// Extracts human body poses from a pixel buffer.
    ///
    /// - Parameters:
    ///   - image: An image.

```

```

    /// - eventHandler: An event handler.
    /// - Returns: A array of poses from all detected persons.
    public func applied CIImage
EventHandler nil async throws Pose

    /// The input type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias Input CIImage

    /// The output type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias Output Pose

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanBodyPoseExtractor
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`-
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

    /// The human hand pose image feature extractor.
    @available macOS 13.0 iOS 16.0 tvOS 16.0
    public struct HumanHandPoseExtractor Transformer Sendable

    /// Creates a human hand pose extractor transformer

```

```

public init

    /// Extracts human hand poses from a pixel buffer.
    ///
    /// - Parameters:
    ///   - image: An image.
    ///   - eventHandler: An event handler.
    /// - Returns: A array of poses from all detected hands.
    public func applied CIImage
EventHandler nil async throws Pose

    /// The input type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input CIImage

    /// The output type.
    @available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output Pose

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension HumanHandPoseExtractor
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`  

    /// initializer. This initializer works with any type, and uses the custom  

    /// `debugDescription` property for types that conform to  

    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

```

```
/// An image blurring transformer.  
@available macOS 14.0 iOS 17.0 tvOS 17.0  
public struct ImageBlur Transformer Sendable  
  
    /// The radius determines how many pixels are used to create the blur. The  
    larger the radius, the blurrier the result.  
    public var radius Double  
  
    /// Creates a transformer that blurs an image.  
    ///  
    /// - Parameter radius: The radius determines how many pixels are  
    used to create the blur. The larger the radius,  
    /// the blurrier the result.  
    public init Double  
  
    /// Blurs an image using a disc-shaped convolution kernel.  
    ///  
    /// - Parameters:  
    ///   - image: An image.  
    ///   - eventHandler: An event handler.  
    /// - Returns: The blurry image.  
    public func applied CIIImage  
EventHandler nil CIIImage  
  
    /// The input type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Input CIIImage  
  
    /// The output type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Output CIIImage  
  
/// An image color transformer.  
@available macOS 14.0 iOS 17.0 tvOS 17.0  
public struct ImageColorTransformer Transformer Sendable  
  
    /// The brightness adjustment, between 0.0 and 1.0.  
    public var brightness Float  
  
    /// The contrast adjustment, between 0.0 and 1.0.  
    public var contrast Float  
  
    /// The hue adjustment, between 0.0 and 1.0.  
    public var hue Float  
  
    /// The saturation adjustment, between 0.0 and 1.0.  
    public var saturation Float
```

```
/// Creates an image color transformer.  
///  
/// - Parameters:  
///   - brightness: The brightness adjustment, between 0.0 and 1.0.  
///   - contrast: The contrast adjustment, between 0.0 and 1.0.  
///   - hue: The hue adjustment, between 0.0 and 1.0.  
///   - saturation: The saturation adjustment, between 0.0 and 1.0.  
public init Float nil Float  
nil Float nil Float nil  
  
/// Performs the image color transformation operation on the input image.  
///  
/// - Parameters:  
///   - image: An image.  
///   - eventHandler: An event handler.  
/// - Returns: The color transformed image.  
public func applied CIImage  
EventHandler nil CIImage  
  
/// The input type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Input CIImage  
  
/// The output type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Output CIImage  
  
/// An image crop transformer.  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public struct ImageCropper Transformer Sendable  
  
/// The crop rectangle within the image bounds.  
public var cropRectangle CGRect  
  
/// Creates an image crop transformer. This transformer is used to crop an  
image to the `cropRectangle`.  
///  
/// - Parameters:  
///   - cropRectangle: A crop rectangle to use. It must always be within  
the input images' bounds.  
public init CGRect  
  
/// Perform the image crop operation on the input pixelBuffer.  
/// - Parameters:  
///   - image: An image.  
///   - eventHandler: An event handler.  
/// - Returns: The cropped pixel buffer.  
public func applied CIImage
```

```

EventHandler    nil    throws    CIImage

    /// The input type.
    @available iOS 16.0  tvOS 16.0  macOS 13.0
    public typealias Input    CIImage

    /// The output type.
    @available iOS 16.0  tvOS 16.0  macOS 13.0
    public typealias Output   CIImage

@available macOS 13.0  iOS 16.0  tvOS 16.0
extension ImageCropper  CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    ///     The conversion of `p` to a string in the assignment to `s` uses the
    ///     `Point` type's `debugDescription` property.
public var debugDescription  String    get

@available macOS 13.0  iOS 16.0  tvOS 16.0
extension ImageCropper  Codable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.

```

```
public init           any Decoder throws
    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode      any Encoder throws
```

```
/// An image exposure adjusting transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public struct ImageExposureAdjuster : Transformer, Sendable
    /// The amount to adjust the exposure of the image. The larger the value, the
    /// brighter the exposure.
    public var amount : Double
```

```
    /// Creates an image exposure adjusting transformer.
    ///
    /// - Parameter amount: The amount to adjust the exposure of the
    /// image. The larger the value, the brighter the exposure.
```

```
    public init : Double
        /// Adjusts the exposure of the input image.
        ///
        /// - Parameters:
        ///   - image: An image.
        ///   - eventHandler: An event handler.
        /// - Returns: An image with adjusted exposure.
        public func applied : CIImage
EventHandler nil CIImage
```

```
    /// The input type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Input = CIImage
```

```
    /// The output type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Output = CIImage
```

```
    /// A transformer that takes an image and outputs image features.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public protocol ImageFeatureExtractor : Transformer where
Self Input : CIImage, Self Output : MLShapedArray<Float>
```

```

/// ImageFeaturePrint image feature extractor.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct ImageFeaturePrint : ImageFeatureExtractor

    /// The crop and scale options.
    public let cropAndScale : VNImageCropAndScaleOption

    /// The latest feature extractor revision.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public static let latestRevision : Int

    /// The feature extractor revision number.
    ///
    /// There are two Image Feature Print revisions: 1 and 2. In most cases
    revision 2 produces better models because
        /// it uses a smaller feature vector and better features.
    ///
    /// In iOS 12 or later, tvOS 12 or later, and macOS 10.14 or later, revision 1
    takes images with a size of 299x299
        /// and produces a 2048 feature vector.
    ///
    /// In iOS 17 or later, tvOS 17 or later, and macOS 14 or later, revision 2
    takes images with a size of 360x360
        /// and produces a 768 feature vector.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public var revision : Int

    /// Creates a FeaturePrint feature extractor.
    ///
    /// - Parameters:
    ///   - cropAndScale: The scaling and cropping options.
    ///   - context: The CoreImage context to use for the operation.
    Defaults to a new default context.
    public init VNImageCropAndScaleOption
                    CIContext

    /// Creates a FeaturePrint feature extractor.
    ///
    /// - Parameters:
    ///   - revision: The revision of feature extractor to use.
    ///   - cropAndScale: The scaling and cropping options.
    ///   - context: The CoreImage context to use for the operation.
    Defaults to a new default context.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public init Int
                    VNImageCropAndScaleOption
                    CIContext

```

```

    /// Extracts image features from an image.
    ///
    /// - Parameters:
    ///   - image: An image.
    ///   - eventHandler: An event handler.
    /// - Returns: A shaped array containing the extracted features of the
image.
    public func applied           CIImage
EventHandler nil async throws  MLShapedArray Float

    /// The input type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input   CIImage

    /// The output type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output  MLShapedArray Float

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension ImageFeaturePrint  CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`-
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription  String  get

```

`@available macOS 13.0 iOS 16.0 tvOS 16.0`

```
extension ImageFeaturePrint Codable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

/// An image flipper transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public struct ImageFlipper Transformer Sendable

    /// The orientation to flip the image.
public var orientation ImageFlipper Orientation

    /// Creates an image flipper.
    ///
    /// - Parameter orientation: The orientation to flip the image.
public init ImageFlipper Orientation

    /// Performs the image flipper operation on the input image.
    /// - Parameters:
    ///   - image: An image.
    ///   - eventHandler: An event handler.
    /// - Returns: The flipped image.
public func applied CIImage
EventHandler nil CIImage

    /// The input type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Input CIImage

    /// The output type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Output CIImage
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ImageFlipper

    /// The orientation to flip the image.
    public enum Orientation Hashable Sendable

        /// Flip the image horizontally.
        case horizontal

        /// Flip the image vertically.
        case vertical

        /// Returns a Boolean value indicating whether two values are equal.
        ///
        /// Equality is the inverse of inequality. For any values `a` and `b`,
        /// `a == b` implies that `a != b` is `false`.
        ///
        /// - Parameters:
        ///   - lhs: A value to compare.
        ///   - rhs: Another value to compare.
        public static func ImageFlipper Orientation
ImageFlipper Orientation Bool

    /// Hashes the essential components of this value by feeding them into
the /// given hasher.
///
/// Implement this method to conform to the `Hashable` protocol. The
/// components used for hashing must be the same as the components
compared /// compared
/// in your type's `==` operator implementation. Call
`hasher.combine(_:)` /// with each of these components.
///
/// - Important: In your implementation of `hash(into:)`,
/// don't call `finalize()` on the `hasher` instance provided,
/// or replace it with a different instance.
/// Doing so may become a compile-time error in the future.
///
/// - Parameter hasher: The hasher to use when combining the
components /// components
/// of this instance.
public func hash inout Hasher

    /// The hash value.
///
/// Hash values are not guaranteed to be equal across different
executions of
```

```
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)`
requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get
```

```
/// An image file reader.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct ImageReader Transformer Sendable
```

```
/// Creates an image reader.
public init
```

```
/// Reads an image URL as a `CIImage`.
```

```
///

```

```
/// - Parameters:
```

```
/// - url: A image URL.
```

```
/// - eventHandler: An event handler.
```

```
/// - Returns: An image.
```

```
public func applied URL
EventHandler nil throws CIImage
```

```
/// Reads an image URL as a `CIImage`.
```

```
///

```

```
/// - Parameter url: A image URL.
```

```
/// - Returns: An image.
```

```
public static func read URL throws CIImage
```

```
/// The input type.
```

```
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input URL
```

```
/// The output type.
```

```
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output CIImage
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension ImageReader CustomDebugStringConvertible
```

```
/// A textual representation of this instance, suitable for debugging.
```

```
///

```

```
/// Calling this property directly is discouraged. Instead, convert an
```

```

    /// instance of any type to a string by using the `String(reflecting:)`  

    /// initializer. This initializer works with any type, and uses the custom  

    /// `debugDescription` property for types that conform to  

    /// `CustomDebugStringConvertible`:  

    ///  

    ///     struct Point: CustomDebugStringConvertible {  

    ///         let x: Int, y: Int  

    ///  

    ///         var debugDescription: String {  

    ///             return "(\(x), \(y))"  

    ///         }  

    ///     }  

    ///  

    ///     let p = Point(x: 21, y: 30)  

    ///     let s = String(reflecting: p)  

    ///     print(s)  

    ///     // Prints "(21, 30)"  

    ///  

    /// The conversion of `p` to a string in the assignment to `s` uses the  

    /// `Point` type's `debugDescription` property.  

public var debugDescription String get

```

```

    /// An image rotating transformer.  

@available macOS 14.0 iOS 17.0 tvOS 17.0  

public struct ImageRotator Transformer Sendable

    /// The angle, in radians, by which to rotate the coordinate space of the  

    /// specified context. Positive values rotate  

    /// counterclockwise and negative values rotate clockwise.  

public var angle Double

    /// Creates a transformer that rotates an image by a specified angle.  

    ///  

    /// – Parameter angle: The angle, in radians, by which to rotate the  

    /// coordinate space of the specified context.  

    /// Positive values rotate counterclockwise and negative values rotate  

    /// clockwise.  

public init Double

    /// Rotates the image and then scales and crops the rotated image to fit the  

    /// extent of the input image.  

    ///  

    /// – Parameters:  

    ///     – image: An image.  

    ///     – eventHandler: An event handler.  

    /// – Returns: The rotated image.  

public func applied CIIImage  

EventHandler nil CIIImage

```

```
/// The input type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Input CIImage  
  
/// The output type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias Output CIImage  
  
/// An image scaling transformer.  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public struct ImageScaler Transformer Sendable  
  
/// The target image size.  
public var targetSize CGSize  
  
/// Creates an image scaler transformer. This transformer is used to scale an  
image to the `targetSize`.  
///  
/// - Parameters:  
///   - targetSize: The target image size. Both width and height must  
be positive.  
public init CGSize  
  
/// Creates an image scaler transformer that preserves the aspect ratio.  
///  
/// This transformer scales an image to match the `targetWidth` while  
preserving the aspect ratio.  
///  
/// - Parameters:  
///   - targetWidth: The target image width. It must be positive.  
@available macOS 14.0 iOS 17.0 tvOS 17.0  
public init Double  
  
/// Creates an image scaler transformer that preserves the aspect ratio.  
///  
/// This transformer scales an image to match the `targetHeight` while  
preserving the aspect ratio.  
///  
/// - Parameters:  
///   - targetHeight: The target image height. It must be positive.  
@available macOS 14.0 iOS 17.0 tvOS 17.0  
public init Double  
  
/// Perform the image scaler operation on the input pixelBuffer.  
/// - Parameters:  
///   - image: An image.  
///   - eventHandler: An event handler.  
/// - Returns: The scaled pixel buffer.  
public func applied CIImage
```

```

EventHandler    nil    throws    CIImage

    /// The input type.
    @available iOS 16.0  tvOS 16.0  macOS 13.0
    public typealias Input    CIImage

    /// The output type.
    @available iOS 16.0  tvOS 16.0  macOS 13.0
    public typealias Output   CIImage

@available macOS 13.0  iOS 16.0  tvOS 16.0
extension ImageScaler  CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    ///     // The conversion of `p` to a string in the assignment to `s` uses the
    ///     // `Point` type's `debugDescription` property.
public var debugDescription  String    get

@available macOS 13.0  iOS 16.0  tvOS 16.0
extension ImageScaler  Codable

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.

```

```
public init           any Decoder throws
    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode      any Encoder throws
    /// A transformer that replaces missing values with a pre-defined value.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct ImputeTransformer Element Transformer
 Codable where Element Decodable Element Encodable

    /// Impute value used to replace missing values.
public var value Element

    /// Creates an impute transformer.
    ///
    /// - Parameter value: The value used to replace missing values.
public init Element

    /// Imputes a single input.
    ///
    /// - Parameters:
    ///   - input: The input.
    ///   - eventHandler: An event handler.
@inlinable public func applied Element
    EventHandler nil Element

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Input Element

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Output Element

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
```

```


    /**
     * This function throws an error if any values are invalid for the given
     * encoder's format.
     */
    /**
     * - Parameter encoder: The encoder to write data to.
    public func encode any Encoder throws

     * Creates a new instance by decoding from the given decoder.
     */
    /**
     * This initializer throws an error if reading from the decoder fails, or
     * if the data read is corrupted or otherwise invalid.
     */
    /**
     * - Parameter decoder: The decoder to read data from.
    public init any Decoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ImputeTransformer CustomDebugStringConvertible

     * A textual representation of this instance, suitable for debugging.
     */
    /**
     * Calling this property directly is discouraged. Instead, convert an
     * instance of any type to a string by using the `String(reflecting:)` initializer.
     * This initializer works with any type, and uses the custom
     * `debugDescription` property for types that conform to
     * `CustomDebugStringConvertible`:
     */
    struct Point: CustomDebugStringConvertible {
        let x: Int, y: Int

        var debugDescription: String {
            return "(\(x), \(y))"
        }
    }

    let p = Point(x: 21, y: 30)
    let s = String(reflecting: p)
    print(s)
    // Prints "(21, 30)"

     * The conversion of `p` to a string in the assignment to `s` uses the
     * `Point` type's `debugDescription` property.
    public var debugDescription String get


```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ImputeTransformer Equatable where Element
Equatable

```

```

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func ImputeTransformer Element
ImputeTransformer Element Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension ImputeTransformer Hashable where Element
Hashable

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    /// compared
    /// in your type's `==` operator implementation. Call
    `hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,  

    ///   don't call `finalize()` on the `hasher` instance provided,  

    ///   or replace it with a different instance.  

    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
    /// components
    /// of this instance.
public func hash inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`  

    requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
    instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

```
extension ImputeTransformer Sendable where Element  
Sendable

/// A key that uniquely identifies a joint.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct JointKey Hashable RawRepresentable Sendable

/// The corresponding value of the raw type.
///
/// A new instance initialized with `rawValue` will be equivalent to this
/// instance. For example:
///
///     enum PaperSize: String {
///         case A4, A5, Letter, Legal
///     }
///
///     let selectedSize = PaperSize.Letter
///     print(selectedSize.rawValue)
///     // Prints "Letter"
///
///     print(selectedSize == PaperSize(rawValue:
selectedSize.rawValue))
///     // Prints "true"
public var rawValue String

/// Creates a new instance with the specified raw value.
///
/// If there is no value of the type that corresponds with the specified raw
/// value, this initializer returns `nil`. For example:
///
///     enum PaperSize: String {
///         case A4, A5, Letter, Legal
///     }
///
///     print(PaperSize(rawValue: "Legal"))
///     // Prints "Optional("PaperSize.Legal")"
///
///     print(PaperSize(rawValue: "Tabloid"))
///     // Prints "nil"
///
/// - Parameter rawValue: The raw value to use for the new instance.
public init String

/// A key associated with left ear joint in a body pose.
public static let leftEar JointKey

/// A key associated with left eye joint in a body pose.
```

```
public static let leftEye JointKey  
    /// A key associated with right ear joint in a body pose.  
public static let rightEar JointKey  
    /// A key associated with right eye joint in a body pose.  
public static let rightEye JointKey  
    /// A key associated with neck joint in a body pose.  
public static let neck JointKey  
    /// A key associated with nose joint in a body pose.  
public static let nose JointKey  
    /// A key associated with left shoulder joint in a body pose.  
public static let leftShoulder JointKey  
    /// A key associated with left elbow joint in a body pose.  
public static let leftElbow JointKey  
    /// A key associated with left wrist joint in a body pose.  
public static let leftWrist JointKey  
    /// A key associated with right shoulder joint in a body pose.  
public static let rightShoulder JointKey  
    /// A key associated with right elbow joint in a body pose.  
public static let rightElbow JointKey  
    /// A key associated with right wrist joint in a body pose.  
public static let rightWrist JointKey  
    /// A key associated with root joint in a body pose.  
public static let root JointKey  
    /// A key associated with left hip joint in a body pose.  
public static let leftHip JointKey  
    /// A key associated with left knee joint in a body pose.  
public static let leftKnee JointKey  
    /// A key associated with left ankle joint in a body pose.  
public static let leftAnkle JointKey  
    /// A key associated with right hip joint in a body pose.  
public static let rightHip JointKey  
    /// A key associated with right knee joint in a body pose.  
public static let rightKnee JointKey
```

```
/// A key associated with right ankle joint in a body pose.  
public static let rightAnkle JointKey  
  
/// A key associated with thumb tip joint in a hand pose.  
public static let thumbTip JointKey  
  
/// A key associated with thumb interphalangeal (IP) joint in a hand pose.  
public static let thumbIP JointKey  
  
/// A key associated with thumb metacarpophalangeal (MP) joint in a hand pose.  
public static let thumbMP JointKey  
  
/// A key associated with thumb carpometacarpal (CMC) joint in a hand pose.  
public static let thumbCMC JointKey  
  
/// A key associated with index finger tip joint in a hand pose.  
public static let indexTip JointKey  
  
/// A key associated with index finger's distal interphalangeal (DIP) joint in a hand pose.  
public static let indexDIP JointKey  
  
/// A key associated with index finger's proximal interphalangeal (PIP) joint in a hand pose.  
public static let indexPIP JointKey  
  
/// A key associated with index finger's metacarpophalangeal (MCP) joint in a hand pose.  
public static let indexMCP JointKey  
  
/// A key associated with middle finger tip joint in a hand pose.  
public static let middleTip JointKey  
  
/// A key associated with middle finger's distal interphalangeal (DIP) joint in a hand pose.  
public static let middleDIP JointKey  
  
/// A key associated with middle finger's proximal interphalangeal (PIP) joint in a hand pose.  
public static let middlePIP JointKey  
  
/// A key associated with middle finger's metacarpophalangeal (MCP) joint in a hand pose.  
public static let middleMCP JointKey  
  
/// A key associated with ring finger tip joint in a hand pose.  
public static let ringTip JointKey
```

```
    /// A key associated with ring finger's distal interphalangeal (DIP) joint in a
    hand pose.
    public static let ringDIP JointKey

    /// A key associated with ring finger's proximal interphalangeal (PIP) joint in a
    hand pose.
    public static let ringPIP JointKey

    /// A key associated with ring finger's metacarpophalangeal (MCP) joint in a
    hand pose.
    public static let ringMCP JointKey

    /// A key associated with ring finger tip joint in a hand pose.
    public static let littleTip JointKey

    /// A key associated with ring finger's distal interphalangeal (DIP) joint in a
    hand pose.
    public static let littleDIP JointKey

    /// A key associated with ring finger's proximal interphalangeal (PIP) joint in a
    hand pose.
    public static let littlePIP JointKey

    /// A key associated with ring finger's metacarpophalangeal (MCP) joint in a
    hand pose.
    public static let littleMCP JointKey

    /// A key associated with hand wrist joint in a hand pose.
    public static let wrist JointKey

    /// The raw type that can be used to represent all values of the conforming
    /// type.
    ///
    /// Every distinct value of the conforming type has a corresponding unique
    /// value of the `RawValue` type, but there may be values of the
    `RawValue`
    /// type that don't have a corresponding value of the conforming type.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 macOS 13.0
    public typealias RawValue String

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension JointKey : CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
```

```

/// `CustomDebugStringConvertible`:
///
/// struct Point: CustomDebugStringConvertible {
///     let x: Int, y: Int
///
///     var debugDescription: String {
///         return "(\(x), \(y))"
///     }
/// }
///
/// let p = Point(x: 21, y: 30)
/// let s = String(reflecting: p)
/// print(s)
/// // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

```

/// A joint in a pose that contains a location and scoring information.

@available macOS 13.0 iOS 16.0 tvOS 16.0

public struct JointPoint **Sendable**

/// The key name for the joint
public let **key** **JointKey**

/// The location of the joint point
public var **location** **CGPoint**

/// A detection confidence of the joint
public var **confidence** **Float**

/// Creates a joint point with its key, location and confidence.

///

/// - **Parameters:**

/// - **key**: joint point identifier name.

/// - **location**: a point indicating the location of the joint.

/// - **confidence**: the detection confidence for the joint.

public init **_** **JointKey** **CGPoint**
Float

@available macOS 14.0 iOS 17.0 tvOS 17.0

extension **JointPoint** **Equatable**

/// Returns a Boolean value indicating whether two values are equal.

///

/// Equality is the inverse of inequality. For any values `a` and `b`,

```

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func JointPoint JointPoint
Bool

    /// Joints selector from a pose.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct JointsSelector Transformer Sendable

    /// A list of joint keys to be ignored.
    public var ignoredJoints JointKey

    /// A list of joint keys to be selected.
    public var selectedJoints JointKey

    /// Creates a joint selector transformer using a list of joint keys to be ignored.
    ///
    /// - Parameters:
    ///   - ignoredJoints: joint keys to be ignored and set to zero in the
pose.
    public init JointKey

    /// Creates a joint selector transformer using a list of joint keys to be
selected.
    ///
    /// - Parameters:
    ///   - selectedJoints: joint keys to be selected from the pose.
    public init JointKey

    /// Select joints to be included in the pose. Ignored joints will be reset to zero
in all fields.
    ///
    /// - Parameters:
    ///   - input: A pose.
    ///   - eventHandler: An event handler.
    /// - Returns: A pose with the ignored joints set to zero.
    public func applied Pose
EventHandler nil Pose

    /// The input type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input Pose

    /// The output type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output Pose

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0
extension JointsSelector : CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    ///     // The conversion of `p` to a string in the assignment to `s` uses the
    ///     // `Point` type's `debugDescription` property.
public var debugDescription: String { get }

/// A linear regressor.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct LinearRegressor: SupervisedEstimator
where Scalar: MLShapedArrayScalar, Scalar: BinaryFloatingPoint

    /// The transformer type created by this estimator.
    public typealias Transformer = LinearRegressorModel

    /// The annotation type.
    public typealias Annotation = Scalar

    /// The linear regressor configuration.
    public var configuration: LinearRegressor.Configuration

    /// Creates a linear regressor.

```

```

    /**
     * - Parameter configuration: The configuration.
     */
    public init
    LinearRegressor Scalar Configuration

    /**
     * Fits a linear regressor model to a sequence of examples.
     */
    /**
     * - Parameters:
     *   - input: A sequence of examples used for fitting the regressor.
     *   - eventHandler: An event handler. This method reports maximum error and root-mean-square error metrics.
     */
    /**
     * Returns: The fitted linear regressor model.
     */
    public func fitted Input Input
    EventHandler nil async throws
    LinearRegressorModel Scalar where Input Sequence
    Input Element AnnotatedFeature MLShapedArray Scalar
    Scalar

    /**
     * Fits a linear regressor model to a sequence of examples.
     */
    /**
     * - Parameters:
     *   - input: A sequence of examples used for fitting the regressor.
     *   - validation: A sequence of examples used for validating the fitted regressor.
     *   - eventHandler: An event handler. This method reports maximum error and root-mean-square error metrics.
     */
    /**
     * Returns: The fitted linear regressor model.
     */
    public func fitted Input Validation Input
    Validation Validation EventHandler
    nil async throws LinearRegressorModel Scalar where
    Input Sequence Validation Sequence Input Element
    AnnotatedFeature MLShapedArray Scalar Scalar
    Validation Element AnnotatedFeature MLShapedArray Scalar
    Scalar

    /**
     * Encodes a fitted transformer.
     */
    public func encode _
    LinearRegressorModel Scalar inout any
    EstimatorEncoder throws

    /**
     * Decodes a previously fitted transformer.
     */
    public func decode inout any
    EstimatorDecoder throws LinearRegressorModel Scalar

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension LinearRegressor UpdatableSupervisedEstimator

    /**
     * Creates a default-initialized transformer suitable for incremental fitting.
     */
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

```

```
public func makeTransformer  
LinearRegressorModel Scalar  
  
    /// Updates a transformer with a new sequence of examples.  
    ///  
    /// - Parameters:  
    ///   - transformer: A transformer to update.  
    ///   - input: A sequence of examples.  
    ///   - eventHandler: An event handler.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func update InputSequence _ inout  
LinearRegressor Scalar Transformer  
InputSequence EventHandler async throws where  
InputSequence Sequence InputSequence Element  
AnnotatedFeature MLShapedArray Scalar Scalar  
  
    /// Encodes the transformer and optimizer to an encoder.  
    ///  
    /// - Parameters:  
    ///   - transformer: A transformer this estimator creates.  
    ///   - encoder: An encoder.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func encodeWithOptimizer _ inout any  
LinearRegressorModel Scalar EstimatorEncoder throws  
  
    /// Reads the encoded transformer and optimizer with a decoder.  
    ///  
    /// - Parameter decoder: A decoder.  
    /// - Returns: The decoded transformer.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public func decodeWithOptimizer inout any  
EstimatorDecoder throws LinearRegressorModel Scalar  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension LinearRegressor Sendable where Scalar Sendable  
  
extension LinearRegressor  
  
    /// A linear regressor configuration.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
  
public struct Configuration Hashable Codable Sendable
```

```
    /// Weight of the L2 regularization term.  
    ///  
    /// The larger this weight, the more the model coefficients shrink toward  
0. This introduces bias into the model  
    /// but decreases variance, potentially leading to better predictions. The  
default value is 0.01; setting this  
    /// parameter to 0 corresponds to unregularized logistic regression.  
public var l2Penalty Double  
  
    /// Weight of the L1 regularization term.  
    ///  
    /// Like the L2 penalty, the higher the L1 penalty, the more the estimated  
coefficients shrink toward 0. The L1  
    /// penalty, however, completely zeros out sufficiently small coefficients,  
automatically indicating features that  
    /// are not useful for the model. The default weight of 0 prevents any  
features from being discarded.  
public var l1Penalty Double  
  
    /// The maximum number of allowed passes through the data.  
    ///  
    /// More passes over the data can result in a more accurately trained  
model. Consider increasing this if the  
    /// training accuracy is low. Defaults to 25.  
    ///  
    /// - Note: This parameter is only used by the `fitted` method.  
When using the `update` method it's up to you  
    /// to decide when to stop.  
public var maximumIterations Int  
  
    /// The starting step size to use for the solver.  
    ///  
    /// Defaults to 1.0. If the first iteration takes a considerable amount of  
time, reducing this parameter may speed  
    /// up model training.  
public var stepSize Double  
  
    /// The convergence threshold.  
    ///  
    /// When the residual is within the convergence threshold of the  
objective, training stops. The threshold is  
    /// also used by the `fitted` method to decide when progress is no  
longer being made, in which case the  
    /// training process will stop before convergence and before the  
specified maximum number of iterations (known  
    /// as early stopping).  
    ///  
    /// Consider reducing this value for a more accurately trained model.  
But beware of overfitting if the it is
```

```
    /// set to a very low value. Defaults to 0.01.
    public var convergenceThreshold Double

    /// The number of iterations to use when evaluating whether to stop
early.
    ///
    /// The `fitted` method will stop if no significant progress is made
for this many iterations. Significant
    /// progress happens when the validation error decreases by at least
`convergenceThreshold`.
    ///
    /// - Note: Early stopping only happens when using the `fitted`
method with validation data.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS
11.0
    public var earlyStopIterationCount Int

    /// A Boolean value indicating whether to scale the input features.
    ///
    /// Scaling the features reduces numerical errors.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS
11.0
    public var scaleFeatures Bool

    /// The optimization strategy.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS
11.0
    public var optimizationStrategy OptimizationStrategy

    /// Creates a default linear regressor configuration.
public init

    /// Hashes the essential components of this value by feeding them into
the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,
    /// don't call `finalize()` on the `hasher` instance provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
```

```

    /// of this instance.
public func hash           inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
LinearRegressor Scalar Configuration
LinearRegressor Scalar Configuration     Bool

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an
empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode      any Encoder throws

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init          any Decoder throws

```

```

/// A trained linear regressor model.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct LinearRegressorModel Scalar Regressor where
Scalar : MLShapedArrayScalar, Scalar : BinaryFloatingPoint

/// The input type.
public typealias Input : MLShapedArray<Scalar>

/// The number of features expected in the input.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var featureCount : Int get

/// The linear coefficients.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var coefficients : Scalar get

/// Creates a linear regression model.
///
/// - Parameter coefficients: The coefficients for the linear
regressor.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public init some Sequence<Scalar>

/// Performs a regression on a single input.
///
/// - Parameters:
///   - input: The regressor input.
///   - eventHandler: An event handler.
/// - Returns: A regression.
public func applied : MLShapedArray<Scalar>
EventHandler nil async throws : Scalar

/// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Output : Scalar

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension LinearRegressorModel : Sendable where Scalar : Sendable

/// A time-series forecasting estimator.

```

```

///-
/// - Note: Only `Float` and `Double` are currently supported as the Scalar
type. You may get faster training
///   when using `Float`.
@available macos 15.0  ios 18.0  tvos 18.0  visionos 2.0
watchos 11.0
public struct LinearTimeSeriesForecaster<Scalar>
    : SupervisedEstimator<Sendable where Scalar : MLShapedArrayScalar>, Scalar, BinaryFloatingPoint

    /// The annotation type.
    public typealias Annotation = MLShapedArray<Scalar>

    /// The transformer type created by this estimator.
    public typealias Transformer = LinearTimeSeriesForecaster<Scalar> Model

    /// The configuration.
    public let configuration: LinearTimeSeriesForecaster<Scalar> Configuration

    /// The number of input samples.
    public var inputWindowSize: Int { get }

    /// The number of predicted samples.
    public var forecastWindowSize: Int { get }

    /// Creates a linear time-series forecaster.
    ///
    /// - Parameter configuration: The configuration parameters.
    public init(configuration: LinearTimeSeriesForecaster<Scalar> Configuration)

    /// Fits a model to a sequence of examples.
    ///
    /// This method uses a sliding window to chunk the input features into
    /// features of ``inputWindowSize`` elements
    /// and annotations of ``forecastWindowSize`` elements. If you want to
    /// use a different windowing strategy, use
    /// ``fitted(toWindows:eventHandler:)``.
    ///
    /// - Parameters:
    ///   - input: A sequence of annotated features. Each feature's shape
    ///     should be `[featureSize]` and each
    ///     annotation's shape should be `[annotationSize]`. This method
    ///     divides the input sequence into windows.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted model.
    public func fitted<Some: Sequence<AnnotatedFeature<MLShapedArray<Scalar>>> some

```

```

MLShapedArray Scalar           EventHandler nil
async throws   LinearTimeSeriesForecaster Scalar Model

    /// Fits a model to a sequence of examples with validation.
    ///
    /// This method uses a sliding window to chunk the input features and
validation features into features of
    /// ``inputWindowSize`` elements and annotations of
``forecastWindowSize`` elements. If you want to use a
    /// different windowing strategy, use
``fitted(toWindows:validateOn:eventHandler:)``.
    ///
    /// - Parameters:
    /// - input: A sequence of annotated features. Each feature's shape
should be `[featureSize]` and each
    /// annotation's shape should be `[annotationSize]`.
    /// - validation: A sequence of annotated validation features. The
feature and annotation shapes should be the
    /// same as the input parameter.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted model.
public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar           EventHandler nil
async throws   LinearTimeSeriesForecaster Scalar Model

    /// Fits a model to a sequence of windows.
    ///
    /// - Parameters:
    /// - input: A sequence of annotated windows. Each window's shape
should be `[inputWindowSize, featureSize]`
    /// and each annotation's shape should be `[forecastWindowSize,
annotationSize]`.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted model.
public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar           EventHandler nil
async throws   LinearTimeSeriesForecaster Scalar Model

    /// Fits a model to a sequence of annotated windows with validation.
    ///
    /// - Parameters:
    /// - input: A sequence of annotated windows. Each window's shape
should be `[inputWindowSize, featureSize]`
    /// and each annotation's shape should be `[forecastWindowSize,
annotationSize]`.
    /// - validation: A sequence of annotated validation windows. The

```

```
feature and annotation shapes should be the
    /// same as the input parameter.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted model.
public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar EventHandler nil
async throws LinearTimeSeriesForecaster Scalar Model

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
extension LinearTimeSeriesForecaster

    /// Encodes a fitted model.
public func encode _
LinearTimeSeriesForecaster Scalar Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted model.
public func decode inout any
EstimatorDecoder throws
LinearTimeSeriesForecaster Scalar Transformer

    /// Encodes the model and optimizer to an encoder.
public func encodeWithOptimizer _
LinearTimeSeriesForecaster Scalar Transformer
inout any EstimatorEncoder throws

    /// Reads the encoded model and optimizer with a decoder.
public func decodeWithOptimizer inout any
EstimatorDecoder throws
LinearTimeSeriesForecaster Scalar Transformer

extension LinearTimeSeriesForecaster

    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
    public typealias Configuration
LinearTimeSeriesForecasterConfiguration

extension LinearTimeSeriesForecaster

    /// A linear time-series forecasting model.
    ///
```

```
    /// - Note: Only `Float` and `Double` are currently supported as the
    Scalar type.
    @available(macOS 15.0, iOS 18.0, tvOS 18.0, visionOS 2.0
watchOS 11.0)
public struct Model : Transformer, Sendable

    /// The input type.
    public typealias Input = MLShapedArray<Scalar>

    /// The output type.
    public typealias Output = MLShapedArray<Scalar>

    /// The number of features per sample.
    public var featureSize: Int { get }

    /// The number of annotations per sample.
    public var annotationSize: Int { get }

    /// The number of input samples.
    public var inputWindowSize: Int { get }

    /// The number of prediction samples.
    public var forecastWindowSize: Int { get }

    /// The number of samples between windows.
    public var stride: Int

    /// The linear coefficients.
    public var weight: MLShapedArray<Scalar> { get }

    /// The bias coefficients.
    public var bias: MLShapedArray<Scalar> { get }

    /// Performs a prediction on a sequence of input features.
    ///
    /// This method uses a sliding window to chunk the input features into
    ``inputWindowSize`` elements every
    ``stride`` elements. If you want to use a different windowing
    strategy, use
    ``applied(toWindow:eventHandler:)``.

    /// - Parameters:
    ///   - input: An sequence of input features. Each feature's shape
    must be ``[featureSize]``. If the sequence
    contains less than ``inputWindowSize`` elements the
    method returns an empty array.
    ///   - eventHandler: An event handler.
    /// - Returns: An array of predictions. Each prediction's shape is
    ``[forecastWindowSize, annotationSize]``.
    ///   The number of predictions depends on the input sequence length
```

and the ``stride`` property.

```
public func applied           some
Sequence MLShapedArray Scalar      EventHandler
nil  async throws   MLShapedArray Scalar

    /// Performs a prediction on a shaped array of features.
    ///
    /// This method uses a sliding window to chunk the input features into
``inputWindowSize`` elements every
    /// ``stride`` elements. If you want to use a different windowing
strategy, use
    /// ``applied(toWindow:eventHandler:)``.

    ///
    /// - Parameters:
    ///   - input: An shaped array of features. The shape must be
`[N, featureSize]` where `N` is the length of the
    ///   sequence, which must be at least `inputWindowSize`.
    ///   - eventHandler: An event handler.
    /// - Returns: A shaped array of predictions with shape `[M,
forecastWindowSize, annotationSize]` where `M`
    ///   is the number of predictions based on the input sequence length
and the ``stride`` property.
public func applied           MLShapedArray Scalar
EventHandler      nil  async throws
MLShapedArray Scalar

    /// Performs a prediction on a window of input features.
    ///
    /// - Parameters:
    ///   - input: An window of input features with shape
`[inputWindowSize, featureSize]`.
    ///   - eventHandler: An event handler.
    /// - Returns: A shaped array of predictions with shape
`[forecastWindowSize, annotationSize]`.
    public func applied
MLShapedArray Scalar           EventHandler      nil
async throws   MLShapedArray Scalar
```

```
@available macOS 15.0  iOS 18.0  tvOS 18.0  visionOS 2.0
watchOS 11.0
```

```
extension LinearTimeSeriesForecaster
UpdatableSupervisedEstimator
```

```
/// Creates a default-initialized model suitable for incremental fitting.
```

```
public func makeTransformer
LinearTimeSeriesForecaster Scalar  Model
```

```

    /// Updates a model with a sequence of features.
    ///
    /// This method uses a sliding window to chunk the input features into
    features of ``inputWindowSize`` elements
    /// and annotations of ``forecastWindowSize`` elements. If you want to
    use a different windowing strategy, use
    /// ``update(_:withWindows:eventHandler:)``.
    ///
    /// - Parameters:
    ///   - model: The model to update.
    ///   - input: A sequence of annotated features. The feature shape must
    be `[featureSize]` and the annotation shape
    ///     must be `[annotationSize]`.
    ///   - eventHandler: An event handler.
public func update _ inout
LinearTimeSeriesForecaster Scalar Model some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar EventHandler nil
async throws

    /// Updates a model with a sequence of windows.
    ///
    /// For faster updates, consider passing a single ``AnnotatedBatch``
    with shaped arrays that contain multiple
    /// training examples. See ``update(_:with:)50gl5``.
    ///
    /// - Parameters:
    ///   - model: The model to update.
    ///   - windows: A sequence of annotated windows. The feature shape
    must be `[inputWindowSize, featureSize]` and
    ///     the annotation shape must be `[forecastWindowSize,
    annotationSize]`.
    ///   - eventHandler: An event handler.
public func update _ inout
LinearTimeSeriesForecaster Scalar Model
some Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar EventHandler nil
async throws

    /// Updates a model with a new batch of examples.
    ///
    /// Use ``TimeSeriesForecasterBatches`` to convert a shaped array
    of features into batches of windowed
    /// features and annotations. Here is an example of training a forecaster:
    ///
    ///
    /// let estimator =
LinearTimeSeriesForecaster<Float>(configuration:
configuration)
    /// var model = estimator.makeTransformer()

```

```

    /**
     /// let batches = try TimeSeriesForecasterBatches(
     ///   features: features,           // shape [N,
featureSize]
     ///   annotations: annotations, // shape [N,
annotationSize]
     ///   batchSize: 32,
     ///   inputWindowSize: configuration.inputWindowSize,
     ///   forecastWindowSize:
configuration.forecastWindowSize,
     ///   shufflesBatches: true
     /// )
     ///
     /// for iteration in 0 ..<
configuration.maximumIterationCount {
    ///   for batch in batches {
    ///     let loss = try await estimator.update(&model,
with: batch)
    ///     print("Loss: \(loss)")
    ///   }
    /// }
    ///
    ///
    /// - Parameters:
    ///   - model: The model to update.
    ///   - input: A shaped array of windowed features. The shape should
be
    ///     `[batchSize, inputWindowSize, featureSize]`.
public func update _ inout
LinearTimeSeriesForecaster Scalar Transformer
AnnotatedBatch Scalar async throws Scalar

```

```

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
extension LinearTimeSeriesForecaster Model
TemporalTransformer

```

```

    /// The output async sequence type.
public typealias OutputSequence
AnyTemporalSequence MLShapedArray Scalar

    /// Performs the transformation on an input sequence.
    ///
    /// - Parameters:
    ///   - input: A temporal sequence of features. Each feature's shape
must be `[featureSize]`.
    ///   - eventHandler: An event handler.
    /// - Returns: An temporal sequence of predictions. Each prediction's
shape is

```

```
    ///`[forecastWindowSize, annotationSize]`.
    public func applied some
TemporalSequence MLShapedArray Scalar
EventHandler nil async throws
AnyTemporalSequence MLShapedArray Scalar

extension LinearTimeSeriesForecaster Model

    /// Exports this transformer as a CoreML model package.
    ///
    /// - Parameter url: The location to write the model into.
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0

    @available
    public func export URL throws

    /// Exports this transformer as a CoreML model package with user-supplied
metadata.
    ///
    /// - Parameters:
    ///   - url: The location to write the model into.
    ///   - metadata: Contextual user-provided information.
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0

    @available
    public func export URL ModelMetadata
throws

    /// The configuration for a linear time-series forecaster.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct LinearTimeSeriesForecasterConfiguration
Hashable Codable Sendable

    /// The number of input samples.
    public var inputWindowSize Int

    /// The number of predicted samples.
    public var forecastWindowSize Int

    /// The number of examples in each training batch.
    ///
    /// - Note: This parameter is only used by the `fitted` method.
    public var batchSize Int

    /// The maximum number of allowed passes through the data.
    ///
```

```
    /// More passes over the data can result in a more accurately trained model.  
    Consider increasing this if the  
    /// training accuracy is low. Defaults to 25.  
    ///  
    /// - Note: This parameter is only used by the `fitted` method. When  
    using the `update` method it's up to you  
    /// to decide when to stop.  
public var maximumIterationCount Int  
  
    /// The early-stopping tolerance.  
    ///  
    /// The tolerance is used by the `fitted` method to decide when progress  
    is no longer being made, in which case the  
    /// training process will stop before the specified maximum number of  
    iterations (known as early stopping).  
    /// Significant progress happens when the validation loss decreases by at  
    least the tolerance.  
    ///  
    /// Defaults to 0.01.  
    ///  
    /// - Note: Early stopping only happens when using the `fitted`  
    method with validation data.  
public var earlyStoppingTolerance Float  
  
    /// The number of iterations to use when evaluating whether to stop early.  
    ///  
    /// The `fitted` method will stop if no significant progress is made for this  
    many iterations. Significant  
    /// progress happens when the validation error decreases by at least  
    `convergenceThreshold`.  
    ///  
    /// - Note: Early stopping only happens when using the `fitted`  
    method with validation data.  
public var earlyStoppingIterationCount Int  
  
    /// The starting learning rate.  
    ///  
    /// Defaults to 0.005.  
public var learningRate Float  
  
    /// A seed to generate reproducible results from random operations.  
public var randomSeed Int  
  
    /// Creates a configuration.  
    ///  
    /// - Parameters:  
    ///   - inputWindowSize: The number of input samples.  
    ///   - forecastWindowSize: The number prediction samples.  
public init Int Int
```

```
    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
    /// in your type's `==` operator implementation. Call
    `hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,  

    ///   don't call `finalize()` on the `hasher` instance provided,  

    ///   or replace it with a different instance.  

    ///   Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
    components
    /// of this instance.
```

public func hash **inout** Hasher

/// Returns a Boolean value indicating whether two values are equal.

///

/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.

///

/// - **Parameters:**

/// - lhs: A value to compare.
/// - rhs: Another value to compare.

public static func

LinearTimeSeriesForecasterConfiguration

LinearTimeSeriesForecasterConfiguration Bool

/// Encodes this value into the given encoder.

///

/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.

///

/// This function throws an error if any values are invalid for the given
/// encoder's format.

///

/// - **Parameter** encoder: The encoder to write data to.

public func encode **any** Encoder **throws**

/// The hash value.

///

/// Hash values are not guaranteed to be equal across different executions of
/// your program. Do not save hash values to use during a future execution.

///

/// - **Important:** `hashValue` is deprecated as a `Hashable`
requirement. To

```

    /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

/// A transformer that runs an input through a scale and offset.
@available macos 13.0 ios 16.0 tvos 16.0 watchos 11.0
public struct LinearTransformer Element Transformer
Hashable Codable where Element BinaryFloatingPoint Element
Decodable Element Encodable

    /// The amount to be scaled.
public var scale Element

    /// The amount to be offset after scaling.
public var offset Element

    /// Creates a linear transformer.
    ///
    /// - Parameters:
    ///   - scale: The amount to be scaled.
    ///   - offset: The amount to be offset after scaling.
public init Element Element

    /// Scales an input.
    ///
    /// - Parameters:
    ///   - input: A floating-point value.
    ///   - eventHandler: An event handler.
    /// - Returns: A scaled value.
@inlinable public func applied Element
EventHandler nil Element

    /// Scales a sequence of inputs.
    ///
    /// - Parameters:
    ///   - input: A sequence of input values.
    ///   - eventHandler: An event handler.
    /// - Returns: An array of scaled values.
@inlinable public func applied S S

```

```

Event Handler      nil      Element where Element
S Element S Sequence

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
        /// in your type's `==` operator implementation. Call
        `hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,
        /// don't call `finalize()` on the `hasher` instance provided,
        /// or replace it with a different instance.
        /// Doing so may become a compile-time error in the future.
        ///
        /// - Parameter hasher: The hasher to use when combining the
    components
        /// of this instance.
public func hash           inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func           LinearTransformer Element
LinearTransformer Element     Bool

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Input   Element

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Output  Element

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given

```

```

    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension LinearTransformer : CustomDebugStringConvertible

```

```

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    /// struct Point: CustomDebugStringConvertible {
    ///     let x: Int, y: Int
    ///
    ///     var debugDescription: String {
    ///         return "(\(x), \(y))"
    ///     }
    /// }
    ///
    /// let p = Point(x: 21, y: 30)
    /// let s = String(reflecting: p)
    /// print(s)
    /// // Prints "(21, 30)"
    ///

```

```

    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension LinearTransformer Sendable where Element
Sendable

    /// A logistic regression classifier.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public struct LogisticRegressionClassifier Scalar Label
    SupervisedEstimator where Scalar MLShapedArrayScalar Scalar
        BinaryFloatingPoint Label Comparable Label Decodable
    Label Encodable Label Hashable

    /// The transformer type created by this estimator.
    public typealias Transformer
    LogisticRegressionClassifierModel Scalar Label

    /// The annotation type.
    public typealias Annotation Label

    /// The logistic regression classifier configuration.
    public var configuration
    LogisticRegressionClassifier Scalar Label Configuration

    /// The set of possible labels.
    public var labels Set Label

    /// Creates a logistic regression classifier.
    ///
    /// - Parameters:
    ///   - labels: The labels used to train the classifier.
    ///   - configuration: The configuration.
    public init Set Label
    LogisticRegressionClassifier Scalar Label Configuration

    /// Fits a logistic regression classifier model to a sequence of examples while
    validating with a validation sequence.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted Input Input
    EventHandler nil async throws

```

```
LogisticRegressionClassifier Scalar Label Transformer where
Input Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Label

    /// Fits a logistic regression classifier model to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the classifier.
    ///   - validation: A sequence of examples used for validating the
    fitted classifier.
    ///   - eventHandler: An event handler. This method reports accuracy
metrics.
    /// - Returns: The fitted logistic regression classifier model.
public func fitted Input Validation Input
                    Validation EventHandler
nil async throws
LogisticRegressionClassifierModel Scalar Label where Input
Sequence Validation Sequence Input Element
AnnotatedFeature MLShapedArray Scalar Label
Validation Element AnnotatedFeature MLShapedArray Scalar
Label

    /// Encodes a fitted transformer.
public func encode _ LogisticRegressionClassifierModel Scalar Label
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
public func decode inout any
EstimatorDecoder throws
LogisticRegressionClassifierModel Scalar Label

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension LogisticRegressionClassifier
UpdatableSupervisedEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func makeTransformer
LogisticRegressionClassifier Scalar Label Transformer

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
```

```

public func update InputSequence _ inout
LogisticRegressionClassifier Scalar Label Transformer
    InputSequence EventHandler async
throws where InputSequence Sequence InputSequence Element
    AnnotatedFeature MLShapedArray Scalar Label

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    /// - transformer: A transformer this estimator creates.
    /// - encoder: An encoder.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func encodeWithOptimizer _ 
LogisticRegressionClassifier Scalar Label Transformer
inout any EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func decodeWithOptimizer inout any
EstimatorDecoder throws
LogisticRegressionClassifier Scalar Label Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension LogisticRegressionClassifier Sendable where Scalar
Sendable Label Sendable

extension LogisticRegressionClassifier

    /// A logistic regression classifier configuration.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public struct Configuration Hashable Codable Sendable

    /// Weight of the L2 regularization term.
    ///
    /// The larger this weight, the more the model coefficients shrink toward
0. This introduces bias into the model
    /// but decreases variance, potentially leading to better predictions. The
default is 0.01.
public var l2Penalty Double

```

```
    /// Weight of the L1 regularization term.  
    ///  
    /// Like the L2 penalty, the higher the L1 penalty, the more the estimated  
    coefficients shrink toward 0. The L1  
    /// penalty, however, completely zeros out sufficiently small coefficients,  
    automatically indicating features that  
    /// are not useful for the model. The default weight of 0 prevents any  
    features from being discarded.  
    public var l1Penalty Double  
  
    /// The maximum number of allowed passes through the data.  
    ///  
    /// More passes over the data can result in a more accurately trained  
    model. Consider increasing this if the  
    /// training accuracy is low. Defaults to 25.  
    ///  
    /// - Note: This parameter is only used by the `fitted` method.  
When using the `update` method it's up to you  
    /// to decide when to stop.  
    public var maximumIterations Int  
  
    /// The starting step size to use for the solver.  
    ///  
    /// Defaults to 1.0. If the first iteration takes a considerable amount of  
    time, reducing this parameter may speed  
    /// up model training.  
    public var stepSize Double  
  
    /// The convergence threshold.  
    ///  
    /// When the residual is within the convergence threshold of the  
    objective, training stops. The threshold is  
    /// also used by the `fitted` method to decide when progress is no  
    longer being made, in which case the  
    /// training process will stop before convergence and before the  
    specified maximum number of iterations (known  
    /// as early stopping).  
    ///  
    /// Consider reducing this value for a more accurately trained model.  
But beware of overfitting if the it is  
    /// set to a very low value. Defaults to 0.01.  
    public var convergenceThreshold Double  
  
    /// The number of iterations to use when evaluating whether to stop  
early.  
    ///  
    /// The `fitted` method will stop if no significant progress is made  
for this many iterations. Significant  
    /// progress happens when the validation accuracy increases by at  
least `convergenceThreshold`.  
    ///
```

```
    /// - Note: Early stopping only happens when using the `fitted`  
method with validation data.  
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS  
11.0)  
    public var earlyStopIterationCount: Int  
  
    /// A Boolean value indicating whether to scale the input features.  
    ///  
    /// Scaling the features reduces numerical errors.  
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS  
11.0)  
    public var scaleFeatures: Bool  
  
    /// The optimization strategy.  
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS  
11.0)  
    public var optimizationStrategy: OptimizationStrategy  
  
    /// Creates a default logistic regression classifier configuration.  
    public init  
  
        /// Hashes the essential components of this value by feeding them into  
the  
        /// given hasher.  
        ///  
        /// Implement this method to conform to the `Hashable` protocol. The  
        /// components used for hashing must be the same as the components  
compared  
        /// in your type's `==` operator implementation. Call  
`hasher.combine(_:)`  
        /// with each of these components.  
        ///  
        /// - Important: In your implementation of `hash(into:)`,  
        /// don't call `finalize()` on the `hasher` instance provided,  
        /// or replace it with a different instance.  
        /// Doing so may become a compile-time error in the future.  
        ///  
        /// - Parameter hasher: The hasher to use when combining the  
components  
        /// of this instance.  
    public func hash(inout Hasher)  
  
        /// Returns a Boolean value indicating whether two values are equal.  
        ///  
        /// Equality is the inverse of inequality. For any values `a` and `b`,  
        /// `a == b` implies that `a != b` is `false`.  
        ///  
        /// - Parameters:  
        ///     - lhs: A value to compare.  
        ///     - rhs: Another value to compare.
```

```

public static func
LogisticRegressionClassifier Scalar Label Configuration
LogisticRegressionClassifier Scalar Label Configuration
Bool

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an
empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode           any Encoder throws

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable` requirement. To
conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init           any Decoder throws

/// A trained logistic regression classifier model.
@available macos 13.0 ios 16.0 tvos 16.0 watchos 11.0
public struct LogisticRegressionClassifierModel Scalar Label
    Classifier where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint Label Comparable Label Decodable
Label Encodable Label Hashable

    /// The input type.

```

```

public typealias Input MLShapedArray Scalar

/// The number of features expected in the input.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var featureCount Int get

/// The linear coefficients.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var coefficients Scalar get

/// Creates a logistic regression model.
///
/// - Parameters:
///   - coefficients: The linear coefficients.
///   - labels: The set of labels.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public init some Sequence Scalar
Set Label

/// Performs a classification on a single input.
///
/// - Parameters:
///   - input: The classifier input.
///   - eventHandler: An event handler.
/// - Returns: A classification distribution.
public func applied MLShapedArray Scalar
Event Handler nil async throws
ClassificationDistribution Label

/// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Output
ClassificationDistribution Label

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension LogisticRegressionClassifierModel Sendable where
Scalar Sendable Label Sendable

/// A transformer that uses a Core ML model as a classifier.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct MLModelClassifierAdaptor Scalar Classifier
where Scalar MLShapedArrayScalar Scalar
BinaryFloatingPoint

```

```

/// The classifier label type.
public enum Label  Hashable  Sendable

    /// The label is string type.
    case string String

    /// The label is integer type.
    case int Int

        /// Hashes the essential components of this value by feeding them into
the
        /// given hasher.
        ///
        /// Implement this method to conform to the `Hashable` protocol. The
        /// components used for hashing must be the same as the components
compared
        /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,  

        /// don't call `finalize()` on the `hasher` instance provided,  

        /// or replace it with a different instance.  

        /// Doing so may become a compile-time error in the future.
        ///
        /// - Parameter hasher: The hasher to use when combining the
components
        /// of this instance.
public func hash           inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,  

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
MLModelClassifierAdaptor Scalar  Label
MLModelClassifierAdaptor Scalar  Label      Bool

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
        /// your program. Do not save hash values to use during a future
execution.
    ///

```

```
    /// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To  
    /// conform to `Hashable`, implement the `hash(into:)`  
requirement instead.  
    /// The compiler provides an implementation for `hashValue` for  
you.  
    public var hashValue Int get  
  
    /// The CoreML model.  
    public let model MLModel  
  
    /// Creates a model adaptor from a CoreML model URL.  
    public init URL  
MLModelConfiguration throws  
  
    /// Creates a MLModel classifier adaptor from a model.  
    /// - Parameters:  
    ///   - model: An MLModel.  
    public init MLModel throws  
  
    /// Performs a prediction from a single input.  
    ///  
    /// - Parameters:  
    ///   - input: The input feature.  
    ///   - eventHandler: An event handler.  
    /// - Returns: A classification distribution.  
    public func applied MLShapedArray Scalar  
        EventHandler nil async throws  
ClassificationDistribution MLModelClassifierAdaptor Scalar La  
bel  
  
    /// The input type.  
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
    public typealias Input MLShapedArray Scalar  
  
    /// The output type.  
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
    public typealias Output  
ClassificationDistribution MLModelClassifierAdaptor Scalar La  
bel  
  
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MLModelClassifierAdaptor Label  
CustomDebugStringConvertible  
  
    /// A text representation of the label.
```

```
public var debugDescription String get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension MLModelClassifierAdaptor Label
ExpressibleByStringLiteral

    /// Creates a Label with a string literal.
    public init String

    /// A type that represents an extended grapheme cluster literal.
    ///
    /// Valid types for `ExtendedGraphemeClusterLiteralType` are
    `Character`,
    /// `String`, and `StaticString`.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias ExtendedGraphemeClusterLiteralType
String

    /// A type that represents a string literal.
    ///
    /// Valid types for `StringLiteralType` are `String` and
    `StaticString`.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias StringLiteralType String

    /// A type that represents a Unicode scalar literal.
    ///
    /// Valid types for `UnicodeScalarLiteralType` are
    `Unicode.Scalar`,
    /// `Character`, `String`, and `StaticString`.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias UnicodeScalarLiteralType String

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension MLModelClassifierAdaptor Label
ExpressibleByIntegerLiteral

    /// Creates a Label with an integer literal.
    public init Int

    /// A type that represents an integer literal.
    ///
    /// The standard library integer and floating-point types are all valid types
    /// for `IntegerLiteralType`.
```

```
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
public typealias IntegerLiteralType Int  
  
/// An image feature extractor provided by an MLModel.  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public struct MLModelImageFeatureExtractor  
ImageFeatureExtractor  
  
/// The CoreML model with .mlmodel extension.  
///  
/// This model should satisfy the following requirements:  
/// 1. Take in one input of type .image and its key name should be same as  
`inputName`.  
/// 2. Any other input should be optional.  
/// 3. Give at least one output of type .multiarray and its key name should be  
same as `outputName`  
public let model MLModel  
  
/// The model's input feature name.  
public let inputName String  
  
/// The model's output feature name.  
public let outputName String  
  
/// Creates an image feature extractor from a CoreML model.  
/// - Parameters:  
///   - model: The CoreML model which will be used for feature  
extraction.  
///   - inputName: The name of the input which the `model` expects.  
///   - outputName: The name of the output from the `model`.  
///   - context: A Core Image context.  
public init MLModel String "image"  
                 String            CIContext                 throws  
  
/// Creates an image feature extractor from a CoreML model URL.  
/// - Parameters:  
///   - url: URL of the .mlmodel file.  
///   - configuration: The model configuration of the CoreML model.  
///   - inputName: The name of the input which the `model` expects.  
///   - outputName: The name of the output from the `model`.  
///   - context: The Core Image context.  
public init MLModelConfiguration String "image"  
                 String            CIContext                 URL                 async throws  
  
/// Uses the CoreML model to create image features from the input pixel  
buffer.  
/// - Parameters:
```

```

    /// - input: An image.
    /// - eventHandler: An event handler.
    /// - Returns: `ImageFeatures` generated by passing the
    `pixelBuffer` through the `model`.
    public func applied           CIImage
EventHandler nil async throws   MLShapedArray Float

    /// The input type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Input   CIImage

    /// The output type.
@available iOS 16.0 tvOS 16.0 macOS 13.0
public typealias Output  MLShapedArray Float

extension MLModelImageFeatureExtractor

    /// CoreML Extraction error.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public enum Error  Error Equatable
CustomDebugStringConvertible

    /// An error indicating that the mlmodel does not take required input.
case invalidInput String

    /// An error indicating that the mlmodel does not produce the required
output.
case invalidOutput String

    /// A text representation of the error.
public var debugDescription String get

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func
MLModelImageFeatureExtractor Error
MLModelImageFeatureExtractor Error      Bool

    /// A transformer that uses a Core ML model as a regressor.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

```
public struct MLModelRegressorAdaptor Scalar      Regressor
where Scalar  MLShapedArrayScalar  Scalar
BinaryFloatingPoint

    /// The CoreML model.
    public let model  MLModel

    /// Creates a model adaptor from a CoreML model URL.
    public init          URL
MLModelConfiguration  throws

    /// Creates a MLModel regressor adaptor from a model.
    /// - Parameters:
    ///   - model: An MLModel.
    public init          MLModel  throws

    /// Performs a prediction from a single input.
    ///
    /// - Parameters:
    ///   - input: The input feature.
    ///   - eventHandler: An event handler.
    /// - Returns: A regression result.
    public func applied          MLShapedArray Scalar
EventHandler        nil  async throws  Double

    /// The input type.
    @available iOS 16.0  tvOS 16.0  watchOS 11.0  macOS 13.0

    public typealias Input  MLShapedArray Scalar

    /// The output type.
    @available iOS 16.0  tvOS 16.0  watchOS 11.0  macOS 13.0

    public typealias Output  Double

    /// A transformer that uses a Core ML model.
    @available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public struct MLModelTransformerAdaptor Scalar      Transformer
where Scalar  MLShapedArrayScalar  Scalar
BinaryFloatingPoint

    /// The CoreML model.
    public let model  MLModel

    /// Creates a model adaptor from a CoreML model URL.
    public init          URL
MLModelConfiguration  throws
```

```

    /// Creates a model adaptor from an MLModel.
    /// - Parameters:
    ///   - model: An MLModel.
    public init MLModel throws

    /// Performs a transformation on a single input.
    ///
    /// - Parameters:
    ///   - input: The input.
    ///   - eventHandler: An event handler.
    /// - Returns: The transformed input.
    public func applied MLShapedArray Scalar
        EventHandler nil async throws
    MLShapedArray Scalar

    /// The input type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Input MLShapedArray Scalar

    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Output MLShapedArray Scalar

    /// An estimator that scales the input values so that the maximum absolute value
    is 1.0.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public struct MaxAbsScaler Element Estimator where
Element BinaryFloatingPoint Element Decodable Element
Encodable

    /// Creates a max abs scaler.
    public init

    /// Fits a max abs scaler to a sequence of elements.
    ///
    /// - Parameters:
    ///   - input: A sequence of elements.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted S S
        EventHandler nil throws
    MaxAbsScaler Element Transformer where Element S Element
S Sequence

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

```
extension MaxAbsScaler Sendable where Element Sendable

extension MaxAbsScaler

    /// An transformer that scales the input values so that the maximum absolute
    value is 1.0.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public struct Transformer : Transformer Hashable

        /// The fitted maximum absolute value.
        public var maximumAbsoluteValue : Element

        /// Creates a max abs scaling transformer.
        ///
        /// - Parameter maximumAbsoluteValue: The fitted maximum
        absolute value.
        public init Element

        /// Scales the input values by `1 / maximumAbsoluteValue`.
        ///
        /// - Parameters:
        /// - input: A floating-point value.
        /// - eventHandler: An event handler.
        /// - Returns: A scaled value.
        @inlinable public func applied Element
            EventHandler nil Element

        /// Hashes the essential components of this value by feeding them into
        the
        /// given hasher.
        ///
        /// Implement this method to conform to the `Hashable` protocol. The
        /// components used for hashing must be the same as the components
        compared
        /// in your type's `==` operator implementation. Call
        `hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,
        /// don't call `finalize()` on the `hasher` instance provided,
        /// or replace it with a different instance.
        /// Doing so may become a compile-time error in the future.
        ///
        /// - Parameter hasher: The hasher to use when combining the
        components
        /// of this instance.
        public func hash inout Hasher
```

```

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
MaxAbsScaler Element Transformer
MaxAbsScaler Element Transformer Bool

    /// The input type.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Input Element

    /// The output type.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Output Element

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get

available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension MaxAbsScaler Transformer
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
instance of any type to a string by using the `String(reflecting:)` initializer.
This initializer works with any type, and uses the custom
`debugDescription` property for types that conform to
`CustomDebugStringConvertible`:

```

```
///  
///     struct Point: CustomDebugStringConvertible {  
///         let x: Int, y: Int  
///  
///         var debugDescription: String {  
///             return "(\(x), \(y))"  
///         }  
///     }  
///  
///     let p = Point(x: 21, y: 30)  
///     let s = String(reflecting: p)  
///     print(s)  
///     // Prints "(21, 30)"  
///  
///     // The conversion of `p` to a string in the assignment to `s` uses the  
///     // `Point` type's `debugDescription` property.  
public var debugDescription String get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MaxAbsScaler Transformer Encodable
```

```
/// Encodes this value into the given encoder.  
///  
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.  
///  
/// This function throws an error if any values are invalid for the given  
/// encoder's format.  
///  
/// - Parameter encoder: The encoder to write data to.  
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MaxAbsScaler Transformer Decodable
```

```
/// Creates a new instance by decoding from the given decoder.  
///  
/// This initializer throws an error if reading from the decoder fails, or  
/// if the data read is corrupted or otherwise invalid.  
///  
/// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MaxAbsScaler Transformer Sendable where Element  
Sendable
```

```
/// A key that uniquely identifies a metric.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct MetricsKey : Hashable, RawRepresentable, Sendable  
  
    /// The corresponding value of the raw type.  
    ///  
    /// A new instance initialized with `rawValue` will be equivalent to this  
    /// instance. For example:  
    ///  
    ///     enum PaperSize: String {  
    ///         case A4, A5, Letter, Legal  
    ///     }  
    ///  
    ///     let selectedSize = PaperSize.Letter  
    ///     print(selectedSize.rawValue)  
    ///     // Prints "Letter"  
    ///  
    ///     print(selectedSize == PaperSize(rawValue:  
selectedSize.rawValue)!)  
    ///     // Prints "true"  
public var rawValue: String  
  
    /// Creates a new instance with the specified raw value.  
    ///  
    /// If there is no value of the type that corresponds with the specified raw  
    /// value, this initializer returns `nil`. For example:  
    ///  
    ///     enum PaperSize: String {  
    ///         case A4, A5, Letter, Legal  
    ///     }  
    ///  
    ///     print(PaperSize(rawValue: "Legal"))  
    ///     // Prints "Optional("PaperSize.Legal")"  
    ///  
    ///     print(PaperSize(rawValue: "Tabloid"))  
    ///     // Prints "nil"  
    ///  
    /// - Parameter rawValue: The raw value to use for the new instance.  
public init rawValue: String  
  
    /// A key associated with a temporal stream source (e.g. a file name).  
public static let source: MetricsKey  
  
    /// A key associated with a training accuracy metric.  
public static let trainingAccuracy: MetricsKey
```

```
    /// A key associated with a validation accuracy metric.  
    public static let validationAccuracy MetricsKey  
  
    /// A key associated with a training loss metric.  
    public static let trainingLoss MetricsKey  
  
    /// A key associated with a validation loss metric.  
    public static let validationLoss MetricsKey  
  
    /// A key associated with a training maximum error metric.  
    public static let trainingMaximumError MetricsKey  
  
    /// A key associated with a validation maximum error metric.  
    public static let validationMaximumError MetricsKey  
  
    /// A key associated with a training error metric.  
    public static let trainingError MetricsKey  
  
    /// A key associated with a validation error metric.  
    public static let validationError MetricsKey  
  
    /// A key associated with a training mean average precision metric.  
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
    public static let trainingMeanAveragePrecision MetricsKey  
  
    /// A key associated with a validation mean average precision metric.  
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
    public static let validationMeanAveragePrecision  
MetricsKey  
  
    /// The raw type that can be used to represent all values of the conforming  
    /// type.  
    ///  
    /// Every distinct value of the conforming type has a corresponding unique  
    /// value of the `RawValue` type, but there may be values of the  
`RawValue`  
    /// type that don't have a corresponding value of the conforming type.  
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
    public typealias RawValue String  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MetricsKey CustomDebugStringConvertible  
  
    /// A textual representation of this instance, suitable for debugging.  
    ///
```

```

    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    /// struct Point: CustomDebugStringConvertible {
    ///     let x: Int, y: Int
    ///
    ///     var debugDescription: String {
    ///         return "(\(x), \(y))"
    ///     }
    /// }
    ///
    /// let p = Point(x: 21, y: 30)
    /// let s = String(reflecting: p)
    /// print(s)
    /// // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
public var debugDescription String get

```

```

    /// An estimator that scales the input values so that they all lie in a closed range.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct MinMaxScaler<Element>: Estimator where
Element: BinaryFloatingPoint, Element: Decodable, Element: Encodable

    /// The desired range of transformed values.
public var range: ClosedRange<Element>

    /// Creates a min max scaler.
    /// - Parameter range: The desired range of transformed values.
public init(range: ClosedRange<Element>, lowerBound: Element = 0, upperBound: Element = 1)

    /// Fits a min max scaler to a sequence of elements.
    ///
    /// - Parameters:
    ///   - input: A sequence of elements.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted<S>(input: Sequence<Element>, eventHandler: EventHandler<Element>? = nil) throws MinMaxScaler<Element>: Transformer where Element: Sequence<Element>, S: Sequence<Element>

```

```


@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension MinMaxScaler : Sendable where Element : Sendable

extension MinMaxScaler

    /// A transformer that scales the input values so that they all lie in a closed
    range.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)

    public struct Transformer : Transformer, Hashable

        /// The desired range of transformed values.
        public var desiredRange : ClosedRange<Element>

        /// The fitted range derived by the estimator when fitting.
        public var fittedRange : ClosedRange<Element>

        /// Creates a minmax scaling transformer.
        ///
        /// - Parameters:
        ///   - desiredRange: The desired range of transformed values.
        ///   - fittedRange: The range derived by the estimator when
fitting.

        public init(_ desiredRange: ClosedRange<Element>, _ fittedRange: ClosedRange<Element>)

        /// Scales the input values so that they all lie in the closed range
        ` [minimum, maximum]`.
        ///
        /// - Parameters:
        ///   - input: A floating-point value.
        ///   - eventHandler: An event handler.
        /// - Returns: A scaled value.
        @inlinable public func applied(_ input: Element, _ eventHandler: EventHandler? = nil) Element

        /// Hashes the essential components of this value by feeding them into
the
        /// given hasher.
        ///
        /// Implement this method to conform to the `Hashable` protocol. The
        /// components used for hashing must be the same as the components
compared
        /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,
        /// don't call `finalize()` on the `hasher` instance provided,


```

```
    /// or replace it with a different instance.  
    /// Doing so may become a compile-time error in the future.  
    ///  
    /// - Parameter hasher: The hasher to use when combining the  
components  
    /// of this instance.  
public func hash inout Hasher  
  
    /// Returns a Boolean value indicating whether two values are equal.  
    ///  
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    ///  
    /// - Parameters:  
    ///   - lhs: A value to compare.  
    ///   - rhs: Another value to compare.  
public static func  
MinMaxScaler Element Transformer  
MinMaxScaler Element Transformer Bool  
  
    /// The input type.  
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS  
13.0  
public typealias Input Element  
  
    /// The output type.  
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS  
13.0  
public typealias Output Element  
  
    /// The hash value.  
    ///  
    /// Hash values are not guaranteed to be equal across different  
executions of  
    /// your program. Do not save hash values to use during a future  
execution.  
    ///  
    /// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To  
    /// conform to `Hashable`, implement the `hash(into:)`  
requirement instead.  
    /// The compiler provides an implementation for `hashValue` for  
you.  
public var hashValue Int get  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MinMaxScaler Transformer  
CustomDebugStringConvertible
```

```

/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer. This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
///     struct Point: CustomDebugStringConvertible {
///         let x: Int, y: Int
///
///         var debugDescription: String {
///             return "(\(x), \(y))"
///         }
///     }
///
///     let p = Point(x: 21, y: 30)
///     let s = String(reflecting: p)
///     print(s)
///     // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension MinMaxScaler Transformer Encodable

```

```

/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension MinMaxScaler Transformer Decodable

```

```

/// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.

```

```
///  
/// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension MinMaxScaler Transformer Sendable where Element  
Sendable  
  
/// Errors related to CoreML model compatibility.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public enum ModelCompatibilityError LocalizedError  
Equatable Sendable  
  
    /// An error that indicates that the label output is missing from the model.  
    case missingLabel  
  
    /// An error that indicates that the label probabilities output is missing from  
    the model.  
    case missingLabelProbabilities  
  
    /// An error that indicates that the regressor model output is missing.  
    case missingPredictedFeature  
  
    /// An error that indicates that the label has the wrong type.  
    case incompatibleLabelType  
  
    /// An error that indicates that the input data has the wrong format.  
    case incompatibleInputDataFormat MLFeatureType  
        MLFeatureType  
  
    /// An error that indicates that the input multi array has the wrong value type.  
    case  
        incompatibleInputMultiArrayType MLMultiArrayDataType  
  
    /// An error that indicates that the output data has the wrong format.  
    case incompatibleOutputDataFormat MLFeatureType  
        MLFeatureType  
  
    /// An error that indicates that the number of model inputs is wrong.  
    case incompatibleInputCount Int Int  
  
    /// An error that indicates that the number of model outputs is wrong.  
    case incompatibleOutputCount Int Int  
  
    /// An error that indicates that the input is missing from the model.  
    case missingInput String
```

```
/// An error that indicates that the output is missing from the model.
case missingOutput           String

/// An error that indicates that the metadata key has the wrong type.
@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0

case incompatibleMetadataKey      String

/// A localized message describing what error occurred.
public var errorDescription  String   get

/// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:
///   - lhs: A value to compare.
///   - rhs: Another value to compare.
public static func      ModelCompatibilityError
ModelCompatibilityError    Bool

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension ModelCompatibilityError
CustomDebugStringConvertible

/// A text representation of the error.
public var debugDescription  String   get

/// User info keys that specify useful information about a model.
@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0
public struct ModelMetadata  Hashable  Sendable

/// A short description of what the model does and/or its purpose.
public var description  String

/// A version number encoded as a string.
public var version   String

/// The author of this model.
public var author    String

/// License information for the model.
public var license   String

/// Creator-defined custom metadata.
public var creatorDefined  String   String
```

```
    /// Creates model metadata.
public init           String   ...           String
...          String   ...          String   ...
String   String

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
        /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,  

        /// don't call `finalize()` on the `hasher` instance provided,  

        /// or replace it with a different instance.  

        /// Doing so may become a compile-time error in the future.
        ///
        /// - Parameter hasher: The hasher to use when combining the
components
        /// of this instance.
public func hash           inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,  

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func           ModelMetadata      ModelMetadata
Bool

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`  

requirement. To
        /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
        /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int   get
```

```

/// An updatable model error.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public enum ModelUpdateError : LocalizedError, Equatable, Sendable

    /// An error that indicates that a default initialized transformer suitable for
    /// fitting cannot perform apply before performing an update.
    case invalidState : String

    /// A localized message describing what error occurred.
    public var errorDescription : String { get }

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func == (lhs: ModelUpdateError, rhs: ModelUpdateError) Bool

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension ModelUpdateError : CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription : String { get }

    /// Multi-label classification metrics.
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
public struct MultiLabelClassificationMetrics : Hashable

    /// The number of examples used to compute the metrics.
    public var exampleCount : Int { get }

    /// The classifier labels.
    public var labels : Set<Label> { get }

    /// A dictionary of label and confidence thresholds.
    public var confidenceThresholds : [Label : Float] { get }

    /// The mean average precision.
    ///
    /// An average precision score summarizes the precision-recall curve for a
    /// label.
    /// The mean average precision is the mean of the average precision scores

```

for all the classification labels.

```
public var meanAveragePrecision  Float    get
```

```
    /// Creates multi-label classification metrics for classifications and ground  
    truth labels.
```

```
    ///
```

```
    /// The classifications and ground truth sequences are matched element by  
    element in the order they are provided.
```

```
    /// Both sequences must have the same number of elements.
```

```
    ///
```

```
    /// - Parameters:
```

```
    ///   - classifications: A sequence of classifications.
```

```
    ///   - groundTruth: A sequence of true labels.
```

```
    ///   - strategy: A label confidence threshold selection strategy.
```

```
    ///   - labels: The set of labels to consider.
```

```
public init      some
```

```
Sequence ClassificationDistribution Label      some
```

```
Sequence Set Label
```

```
MultiLabelClassificationMetrics Label ThresholdSelectionStrat  
egy      Set Label    throws
```

```
    /// Creates multi-label classification metrics for classifications and ground  
    truth labels.
```

```
    ///
```

```
    /// The classifications and ground truth sequences are matched element by  
    element in the order they are provided.
```

```
    /// Both sequences must have the same number of elements.
```

```
    ///
```

```
    /// - Parameters:
```

```
    ///   - classifications: A sequence of classifications.
```

```
    ///   - groundTruth: A sequence of true labels.
```

```
    ///   - strategy: A label confidence threshold selection strategy.
```

```
public init      some
```

```
Sequence ClassificationDistribution Label      some
```

```
Sequence Set Label
```

```
MultiLabelClassificationMetrics Label ThresholdSelectionStrat  
egy    throws
```

```
    /// Creates multi-label classification metrics for classifications and ground  
    truth labels.
```

```
    ///
```

```
    /// - Parameters:
```

```
    ///   - pairs: A sequence of pairs with a classification distribution and a  
    set of labels.
```

```
    ///   - strategy: A label confidence threshold selection strategy.
```

```
    ///   - labels: The set of labels to consider.
```

```
public init _      some Sequence
```

```
ClassificationDistribution Label      Set Label
```

```
MultiLabelClassificationMetrics Label ThresholdSelectionStrat
```

```
egy           Set Label  throws

    /// Creates multi-label classification metrics for classifications and ground
    truth labels.
    ///
    /// - Parameters:
    ///   - pairs: A sequence of classifications and true label pairs.
    ///   - strategy: A label confidence threshold selection strategy.
    public init _           some Sequence
ClassificationDistribution Label           Set Label

MultiLabelClassificationMetrics Label ThresholdSelectionStrat
egy  throws

extension MultiLabelClassificationMetrics

    /// Computes the mean average precision.
    ///
    /// An average precision score summarizes the precision-recall curve for a
    label.
    /// The mean average precision is the mean of the average precision scores
    for all the classification labels.
    ///
    /// - Parameters:
    ///   - classifications: A sequence of multi-label classifications.
    ///   - groundTruth: A sequence of multi-label correct labels.
    ///   - labels: The set of labels to consider.
    /// - Returns: The mean average precision.
    @available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0

    public static func
meanAveragePrecisionScore           some
Sequence ClassificationDistribution Label           some
Sequence Set Label           Set Label           Float

    /// Computes the mean average precision.
    ///
    /// An average precision score summarizes the precision-recall curve for a
    label.
    /// The mean average precision is the mean of the average precision scores
    for all the classification labels.
    ///
    /// - Parameters:
    ///   - classifications: A sequence of multi-label classifications.
    ///   - groundTruth: A sequence of multi-label correct labels.
    /// - Returns: The mean average precision.
    @available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0

    public static func
```

```

meanAveragePrecisionScore           some
Sequence ClassificationDistribution Label      some
Sequence Set Label          Float

    /// Computes the mean average precision.
    ///
    /// An average precision score summarizes the precision-recall curve for a
label.
    /// The mean average precision is the mean of the average precision scores
for all the classification labels.
    ///
    /// - Parameters:
    ///   - pairs: A sequence of classifications and true label pairs.
    ///   - labels: The set of labels to consider.
    /// - Returns: The mean average precision.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public static func meanAveragePrecisionScore _      some
Sequence           ClassificationDistribution Label
Set Label          Set Label          Float

    /// Computes the mean average precision.
    ///
    /// An average precision score summarizes the precision-recall curve for a
label.
    /// The mean average precision is the mean of the average precision scores
for all the classification labels.
    ///
    /// - Parameters:
    ///   - pairs: A sequence of classifications and true label pairs.
    ///   - Returns: The mean average precision.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public static func meanAveragePrecisionScore _      some
Sequence           ClassificationDistribution Label
Set Label          Float

extension MultiLabelClassificationMetrics

    /// Computes the precision score for a class label.
    ///
    /// Precision score is computed as the ratio `truePositive /
(truePositive + falsePositive)`.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The precision score for the given label.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func precisionScore           Label      Float

```

```
    /// Computes the recall score for a class label.  
    ///  
    /// Recall score is computed as the ratio `truePositive /  
(truePositive + falseNegative)`.  
    ///  
    /// - Parameter label: The label to use as true positive.  
    /// - Returns: The recall score for the given label.  
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)  
  
public func recallScore  
    Label  
    Float  
  
extension MultiLabelClassificationMetrics  
  
    /// A strategy for selecting a confidence threshold.  
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)  
  
public enum ThresholdSelectionStrategy : Hashable,  
Equatable  
  
    /// A confidence threshold strategy that balances precision and recall  
equivalently.  
    ///  
    /// Use this strategy to select a threshold for each label to maximize the  
f1-score.  
    case balancedPrecisionAndRecall  
  
    /// A confidence threshold strategy for a specific precision that has at  
least a minimum recall value.  
    ///  
    /// This strategy selects a threshold for each label by searching for the  
specified precision value on the label's  
    /// precision-recall curve. At the precision, the recall must be greater  
than or equal to the minimum recall value,  
    /// otherwise a NaN threshold for the corresponding label is returned.  
    ///  
    /// Use this strategy to reduce the rate of false-positive predictions  
while constraining the false-negative  
    /// predictions.  
    case precision : Float  
        Float  
  
    /// A confidence threshold strategy for a recall precision that has at  
least a minimum precision value.  
    ///  
    /// This strategy selects a threshold for each label by searching for the  
specified recall value on the label's  
    /// precision-recall curve. At the recall, the precision must be greater  
than or equal to the minimum precision  
    /// value, otherwise a NaN threshold for the corresponding label is  
returned.
```

```

    /**
     * Use this strategy to reduce the rate of false-negative predictions
     * while constraining the false-positive
     * predictions.
     case recall Float Float

     * A confidence threshold strategy that uses the specified thresholds
     * for each label.
     case fixed Label Float

     * Hashes the essential components of this value by feeding them into
     * the
     * given hasher.
     /**
     * Implement this method to conform to the `Hashable` protocol. The
     * components used for hashing must be the same as the components
     * compared
     * in your type's `==` operator implementation. Call
     `hasher.combine(_:)`
     * with each of these components.
     /**
     * - Important: In your implementation of `hash(into:)`,  

     * don't call `finalize()` on the `hasher` instance provided,  

     * or replace it with a different instance.  

     * Doing so may become a compile-time error in the future.
     /**
     * - Parameter hasher: The hasher to use when combining the
     * components
     * of this instance.
     public func hash inout Hasher

     * Returns a Boolean value indicating whether two values are equal.
     /**
     * Equality is the inverse of inequality. For any values `a` and `b`,  

     * `a == b` implies that `a != b` is `false`.
     /**
     * - Parameters:
     * - lhs: A value to compare.
     * - rhs: Another value to compare.
     public static func
MultiLabelClassificationMetrics Label ThresholdSelectionStrategy
MultiLabelClassificationMetrics Label ThresholdSelectionStrategy
                                Bool

     * The hash value.
     /**
     * Hash values are not guaranteed to be equal across different
     * executions of
     * your program. Do not save hash values to use during a future

```

execution.

```
///  
/// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To  
/// conform to `Hashable`, implement the `hash(into:)`  
requirement instead.  
/// The compiler provides an implementation for `hashValue` for  
you.  
public var hashValue Int get
```

extension MultiLabelClassificationMetrics

```
/// Returns the number of times a label appeared in the ground truth  
collection.  
///  
/// - Parameter label: The label.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
public func count Label Int  
  
/// Returns the number of times the predicted label matched the true label.  
///  
/// If the label does not have a confidence threshold, the true positive count  
is number of elements in the ground truth  
/// labels collection that contains the label. If the label has a confidence  
threshold NaN, the true positive  
/// count is 0. If the label is not in the known set of labels, the true positive  
count is 0.  
///  
/// - Parameter label: The label to use as true positive.  
/// - Returns: The true positive count.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
public func truePositiveCount Label Int  
  
/// Returns the number of times a label was not in the predicted or ground  
truth collections.  
///  
/// If the label does not have a confidence threshold, the true negative count  
is 0. If the label has a  
/// confidence threshold NaN, the true negative count is the number  
elements in the ground truth collection that  
/// do not contain the label. If the label is not in the known set of labels, the  
true negative count is 0.  
///  
/// - Parameter label: The label to use as true positive.  
/// - Returns: The true negative count.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
```

```

public func trueNegativeCount           Label      Int
    /// Returns the number of times the predicted label did not match the true label.
    ///
    /// If the label does not have a confidence threshold, the false positive count is the number of elements in the
    /// ground truth labels collection that do not contain the label. If the label has a confidence threshold NaN, the
    /// false positive count is 0. If the label is not in the known set of labels, the false positive count is 0.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The false positive count.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func falsePositiveCount          Label      Int
    /// Returns the number of times a true label was not predicted.
    ///
    /// If the label does not have a confidence threshold, the false negative count is 0. If the label has a
    /// confidence threshold NaN, the false negative count is the number of elements in the ground truth labels
    /// collection that do not contain the label. If the label is not in the known set of labels, the false negative count is 0.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The false negative count.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func falseNegativeCount          Label      Int
    /// Computes the F1 score from predicted and ground truth values.
    ///
    /// The balanced F-score, or F1 score, is computed as the harmonic mean of the precision and recall.
    /// If the provided label does not have a confidence threshold, the F1 score is NaN.
    ///
    /// - Parameter label: The label to use as true positive.
    /// - Returns: The F1 score for the given label.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func f1Score                  Label      Float

extension MultiLabelClassificationMetrics

    /// Creates empty multi-label classification metrics.
    ///

```

```

    /// - Parameter confidenceThresholds: A dictionary of label and
confidence thresholds.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public init Label Float

    /// Updates the metrics with more classifications and ground truth labels.
    ///
    /// The classifications and ground truth sequences are matched element by
element in the order they are provided.
    /// Both sequences must have the same number of elements.
    ///
    /// - Parameters:
    ///   - classifications: A collection of classifications.
    ///   - groundTruth: A collection of true labels.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public mutating func add some
Sequence ClassificationDistribution Label some
Sequence Set Label

    /// Updates the metrics with more pairs of classifications and ground truth
labels.
    ///
    /// - Parameter pairs: A sequence of classifications and true label
pairs.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public mutating func add _ some
Sequence ClassificationDistribution Label
Set Label

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension MultiLabelClassificationMetrics Sendable where
Label Sendable

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension MultiLabelClassificationMetrics ThresholdSelectionStrategy
Encodable where Label Encodable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.

```

```

    /**
     * - Parameter encoder: The encoder to write data to.
     */
    public func encode(any Encoder throws

@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
extension MultiLabelClassificationMetrics ThresholdSelectionStrategy
Decodable where Label: Decodable

    /**
     * Creates a new instance by decoding from the given decoder.
     */
    /**
     * This initializer throws an error if reading from the decoder fails, or
     * if the data read is corrupted or otherwise invalid.
     */
    /**
     * - Parameter decoder: The decoder to read data from.
     */
    public init(any Decoder throws

@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)
extension MultiLabelClassificationMetrics ThresholdSelectionStrategy
Sendable where Label: Sendable

    /**
     * A multivariate linear regressor.
     */
    /**
     * Unlike a ``LinearRegressor``, a
     * ``MultivariateLinearRegressor`` supports shaped array outputs with any
     * number of
     * elements. It also provides a wider range of training options better suited for
     * large multi-dimensional regression.
     */
    /**
     * - Note: Only `Float` and `Double` are currently supported as the Scalar
     * type. You may get faster training
     * when using `Float`.
     */
    @available(macOS 15.0, iOS 18.0, tvOS 18.0, visionOS 2.0,
    watchOS 11.0)
    public struct MultivariateLinearRegressor
        : SupervisedEstimator, Sendable where Scalar: MLShapedArrayScalar, Scalar == BinaryFloatingPoint

    /**
     * The transformer type created by this estimator.
     */
    public typealias Transformer = MultivariateLinearRegressor

    /**
     * The feature type.
     */
    public typealias Feature = MLShapedArray

    /**
     * The annotation type.
     */

```

```

public typealias Annotation MLShapedArray Scalar

    /// The linear regressor configuration.
    public var configuration
MultivariateLinearRegressor Scalar Configuration

        /// Creates a multivariate linear regressor.
        ///
        /// - Parameter configuration: The configuration.
        public init
MultivariateLinearRegressor Scalar Configuration

        /// Fits a linear regressor model to a sequence of annotated features.
        ///
        /// - Parameters:
        /// - input: A sequence of examples used for fitting the regressor. For
faster processing, instead of passing a
        /// sequence of shaped arrays, consider passing a single shaped array
containing all the training examples. For
        /// example instead of passing `N` shaped arrays with shape `[M]`,,
pass a single shaped array with shape
        /// `[N, M]`. See
``fitted(to:validateOn:eventHandler:)-82szq``.
        /// - eventHandler: An event handler. This method reports mean
squared errors.
        /// - Returns: The fitted linear regressor model.
        public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar EventHandler nil
async throws MultivariateLinearRegressor Scalar Model

        /// Fits a linear regressor model to a sequence of annotated features.
        ///
        /// - Parameters:
        /// - input: A sequence of examples used for fitting the regressor. For
faster processing, instead of passing a
        /// sequence of shaped arrays, consider passing a single shaped array
containing all the training examples. For
        /// example instead of passing `N` shaped arrays with shape `[M]`,,
pass a single shaped array with shape
        /// `[N, M]`. See
``fitted(to:validateOn:eventHandler:)-82szq``.
        /// - validation: A sequence of examples used for validating the
fitted regressor.
        /// - eventHandler: An event handler. This method reports mean
squared errors.
        /// - Returns: The fitted linear regressor model.
        public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar some

```

```

Sequence AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar           EventHandler nil
async throws MultivariateLinearRegressor Scalar Model

    /// Fits a linear regressor model to shaped arrays of features and annotations.
    ///
    /// - Parameters:
    ///   - input: An annotated batch containing the features and annotations. The last dimension of the features is
    ///     the model's input size and the last dimension of the annotations is
    ///     the model's output size. All the
    ///     leading dimensions of the features must match all leading dimensions of the annotations. For example, the
    ///     feature shape can be ` [N, X]` and the annotation shape can be
    ` [N, Y]` for `N` examples where `X` is the
    ///     input size and `Y` is the output size.
    ///   - validation: An annotated batch containing the validation features and annotations. The last dimension of
    ///     the features must be `inputSize` and the last dimension of the annotations must be `outputSize`. All the
    ///     leading dimensions of the features must match all leading dimensions of the annotations.
    ///   - eventHandler: An event handler. This method reports the mean squared errors.
    /// - Returns: The fitted model.
public func fitted AnnotatedBatch Scalar
                        AnnotatedBatch Scalar
EventHandler nil async throws
MultivariateLinearRegressor Scalar Model

    /// Encodes a fitted transformer.
public func encode _
MultivariateLinearRegressor Scalar Model inout
any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
public func decode inout any
EstimatorDecoder throws
MultivariateLinearRegressor Scalar Model

extension MultivariateLinearRegressor

    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
    public typealias Configuration
MultivariateLinearRegressorConfiguration

```

```

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
extension MultivariateLinearRegressor
UpdatableSupervisedEstimator

    /// Creates a default-initialized model suitable for incremental fitting.
    public func makeTransformer
    MultivariateLinearRegressor Scalar Model

    /// Updates a model with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - model: A model to update.
    ///   - input: A sequence of examples. For faster updates, consider
    passing a single ``AnnotatedBatch`` with
        /// shaped arrays that contain multiple training examples. For example
    instead of passing a sequence of `N`
        /// shaped arrays with shape `[M]`, pass a single shaped array with
    shape `[N, M]`. See also
        /// ``update(_:with:)``.
    /// - eventHandler: An event handler. This method reports the mean
    squared error.
    public func update _ inout
    MultivariateLinearRegressor Scalar Model some
    Sequence AnnotatedFeature MLShapedArray Scalar
    MLShapedArray Scalar EventHandler nil
    async throws

    /// Updates a model with a new shaped array of examples.
    ///
    /// - Parameters:
    ///   - model: A model to update.
    ///   - input: An annotated batch containing the features and
    annotations. The last dimension of the features is
        /// the model's input size and the last dimension of the annotations is
    the model's output size. All the
        /// leading dimensions of the features must match all leading
    dimensions of the annotations. For example, the
        /// feature shape can be `[N, X]` and the annotation shape can be
    `[N, Y]` for `N` examples where `X` is the
        /// input size and `Y` is the output size.
    /// - Returns: The mean squared error for the batch, also known as the
    batch loss.
    public func update _ inout
    MultivariateLinearRegressor Scalar Model
    AnnotatedBatch Scalar async throws Scalar

    /// Encodes the model and optimizer to an encoder.
    ///
    /// - Parameters:

```

```

    /// - model: A model this estimator creates.
    /// - encoder: An encoder.
    public func encodeWithOptimizer _  

MultivariateLinearRegressor Scalar Model inout  

any EstimatorEncoder throws  

  

    /// Reads the encoded model and optimizer with a decoder.
    ///  

    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded model.
    public func decodeWithOptimizer inout any  

EstimatorDecoder throws  

MultivariateLinearRegressor Scalar Model

```

extension MultivariateLinearRegressor

```

    /// A trained multivariate linear regressor model.
    ///  

    /// - Note: Only `Float` and `Double` are currently supported as the
Scalar type.  

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  

watchOS 11.0
    public struct Model : Transformer : Sendable

    /// The input type.
    public typealias Input = MLShapedArray

    /// The input size.
    public var inputSize : Int get

    /// The output size.
    public var outputSize : Int get

    /// The linear coefficients.
    public var weight : MLShapedArray get

    /// The bias coefficients.
    public var bias : MLShapedArray get

    /// Creates a multivariate linear regressor.
    ///  

    /// - Parameters:  

    ///   - weight: A shaped array of linear weights.  

    ///   - bias: A one-dimensional shaped array of biases.
    public init weight : MLShapedArray  

bias : MLShapedArray

    /// Performs a prediction on a shaped array.

```

```

    /**
     * - Parameters:
     *   - input: A shaped array of features. The last dimension must
     *     be `inputSize`.
     *   - eventHandler: An event handler.
     * - Returns: A shaped array of predictions. The shape of the
     *     predictions matches the shape of the input
     *     except for the last dimension, which is `outputSize`.
     */
    public func applied<T: MLShapedArray> Scalar<T>
        EventHandler<T> nil async throws
    MLShapedArray<T>

        /**
         * The output type.
         */
        @available(iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0, macOS 15.0)
        public typealias Output = MLShapedArray<T>

```

```

    /**
     * A linear regressor configuration.
     */
    @available(macOS 15.0, iOS 18.0, tvOS 18.0, visionOS 2.0, watchOS 11.0)
    public struct MultivariateLinearRegressorConfiguration<T: Hashable, Codable, Sendable>
        /**
         * The number of examples in each training batch.
         */
        /**
         * - Note: This parameter is only used by the `fitted` method.
         */
        public var batchSize: Int

        /**
         * The maximum number of allowed passes through the data.
         */
        /**
         * More passes over the data can result in a more accurately trained model.
         * Consider increasing this if the
         * training accuracy is low. Defaults to 25.
         */
        /**
         * - Note: This parameter is only used by the `fitted` method. When
         * using the `update` method it's up to you
         * to decide when to stop.
         */
        public var maximumIterationCount: Int

        /**
         * The early-stopping tolerance.
         */
        /**
         * The tolerance is used by the `fitted` method to decide when progress
         * is no longer being made, in which case the
         * training process will stop before the specified maximum number of
         * iterations (known as early stopping).
         */
        /**
         * Significant progress happens when the validation loss decreases by at
         * least the tolerance.
         */

```

```
    /// Defaults to 0.01.  
    ///  
    /// - Note: Early stopping only happens when using the `fitted`  
method with validation data.  
    public var earlyStoppingTolerance Float  
  
    /// The number of iterations to use when evaluating whether to stop early.  
    ///  
    /// The `fitted` method will stop if no significant progress is made for this  
many iterations. Significant  
    /// progress happens when the validation error decreases by at least  
`convergenceThreshold`.  
    ///  
    /// - Note: Early stopping only happens when using the `fitted`  
method with validation data.  
    public var earlyStoppingIterationCount Int  
  
    /// The optimizer learning rate.  
    ///  
    /// Defaults to 0.005.  
    public var learningRate Float  
  
    /// A seed to generate reproducible results from random operations.  
    public var randomSeed Int  
  
    /// Creates a default linear regressor configuration.  
    public init  
  
    /// Hashes the essential components of this value by feeding them into the  
    /// given hasher.  
    ///  
    /// Implement this method to conform to the `Hashable` protocol. The  
    /// components used for hashing must be the same as the components  
compared  
    /// in your type's `==` operator implementation. Call  
`hasher.combine(_:)`  
    /// with each of these components.  
    ///  
    /// - Important: In your implementation of `hash(into:)`,  
    /// don't call `finalize()` on the `hasher` instance provided,  
    /// or replace it with a different instance.  
    /// Doing so may become a compile-time error in the future.  
    ///  
    /// - Parameter hasher: The hasher to use when combining the  
components  
    /// of this instance.  
    public func hash inout Hasher  
  
    /// Returns a Boolean value indicating whether two values are equal.  
    ///
```

```
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    /// - Parameters:  
    ///   - lhs: A value to compare.  
    ///   - rhs: Another value to compare.  
    public static func  
        MultivariateLinearRegressorConfiguration  
        MultivariateLinearRegressorConfiguration      Bool  
  
    /// Encodes this value into the given encoder.  
    /// If the value fails to encode anything, `encoder` will encode an empty  
    /// keyed container in its place.  
    /// This function throws an error if any values are invalid for the given  
    /// encoder's format.  
    /// - Parameter encoder: The encoder to write data to.  
    public func encode      any Encoder  throws  
  
    /// The hash value.  
    /// Hash values are not guaranteed to be equal across different executions of  
    /// your program. Do not save hash values to use during a future execution.  
    /// - Important: `hashValue` is deprecated as a `Hashable`  
    requirement. To  
    /// conform to `Hashable`, implement the `hash(into:)` requirement  
    instead.  
    /// The compiler provides an implementation for `hashValue` for you.  
    public var hashValue  Int  get  
  
    /// Creates a new instance by decoding from the given decoder.  
    /// This initializer throws an error if reading from the decoder fails, or  
    /// if the data read is corrupted or otherwise invalid.  
    /// - Parameter decoder: The decoder to read data from.  
    public init      any Decoder  throws  
  
    /// An estimator that normalizes the input values using a normalization strategy.  
    @available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0  
    public struct NormalizationScaler Element  Estimator where  
        Element: BinaryFloatingPoint, Element: Decodable, Element: Encodable  
  
    /// A normalization strategy.  
    public enum NormalizationStrategy  Sendable
```

```
    /// A normalization strategy that scales by the L1 Norm (sum of vector  
absolute values).
```

```
    case l1
```

```
    /// A normalization strategy that scales by the L2 Norm (Euclidean  
norm).
```

```
    case l2
```

```
    /// Returns a Boolean value indicating whether two values are equal.
```

```
    //
```

```
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.
```

```
    //
```

```
    /// - Parameters:
```

```
    ///   - lhs: A value to compare.
```

```
    ///   - rhs: Another value to compare.
```

```
public static func
```

```
NormalizationScaler Element NormalizationStrategy
```

```
NormalizationScaler Element NormalizationStrategy Bool
```

```
    /// Hashes the essential components of this value by feeding them into  
the
```

```
    /// given hasher.
```

```
    //
```

```
    /// Implement this method to conform to the `Hashable` protocol. The  
    /// components used for hashing must be the same as the components  
compared
```

```
    /// in your type's `==` operator implementation. Call  
`hasher.combine(_:)`
```

```
    /// with each of these components.
```

```
    //
```

```
    /// - Important: In your implementation of `hash(into:)`,  
    /// don't call `finalize()` on the `hasher` instance provided,  
    /// or replace it with a different instance.
```

```
    /// Doing so may become a compile-time error in the future.
```

```
    //
```

```
    /// - Parameter hasher: The hasher to use when combining the  
components
```

```
    /// of this instance.
```

```
public func hash           inout Hasher
```

```
    /// The hash value.
```

```
    //
```

```
    /// Hash values are not guaranteed to be equal across different  
executions of
```

```
    /// your program. Do not save hash values to use during a future  
execution.
```

```
    //
```

```
    /// - Important: `hashValue` is deprecated as a `Hashable`
```

requirement. To

 /// conform to `Hashable`, implement the `hash(into:)` requirement instead.

 /// The compiler provides an implementation for `hashValue` for you.

```
public var hashValue Int get
```

 /// The normalization strategy.

```
public var norm
```

NormalizationScaler Element NormalizationStrategy

 /// Creates a normalization scaler.

 /// - Parameter norm: A selected NormalizationStrategy to scale by.

Defaults as `l2`.

```
public init
```

NormalizationScaler Element NormalizationStrategy

 /// Fits a normalization scaler to a sequence of elements.

 ///

 /// - Parameters:

 /// - input: A sequence of elements.

 /// - eventHandler: An event handler.

 /// - Returns: The fitted transformer.

```
public func fitted S S
```

EventHandler nil throws

NormalizationScaler Element Transformer where Element

S Element S Sequence

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

extension NormalizationScaler Sendable where Element

Sendable

extension NormalizationScaler

 /// A transformer that scales the input using a normalization strategy.

 @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```
public struct Transformer : Transformer, Hashable
```

 /// The normalization scale used to scale the input.

```
public var scale Element
```

 /// Creates a normalization scaling transformer.

 ///

 /// - Parameter scale: The normalization scale.

```
public init Element
```

```

    /// Scales the input values using the calculation `input / scale`.
    ///
    /// - Parameters:
    ///   - input: A floating-point value.
    ///   - eventHandler: An event handler.
    /// - Returns: A scaled value.
@inlinable public func applied Element
EventHandler nil Element

    /// Hashes the essential components of this value by feeding them into
the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,  

    ///   don't call `finalize()` on the `hasher` instance provided,  

    ///   or replace it with a different instance.  

    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,  

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
NormalizationScaler Element Transformer
NormalizationScaler Element Transformer Bool

    /// The input type.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Input Element

    /// The output type.
available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

```

```
public typealias Output Element

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.

public var hashValue Int get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler NormalizationStrategy
Equatable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler NormalizationStrategy Hashable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler Transformer
CustomDebugStringConvertible
```

```
/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer. This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
/// struct Point: CustomDebugStringConvertible {
///     let x: Int, y: Int
///
///     var debugDescription: String {
///         return "(\(x), \(y))"
///     }
/// }
///
/// let p = Point(x: 21, y: 30)
```

```
///      let s = String(reflecting: p)
///      print(s)
///      // Prints "(21, 30)"
///
///      /// The conversion of `p` to a string in the assignment to `s` uses the
///      /// `Point` type's `debugDescription` property.
public var debugDescription String get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler Transformer Encodable
```

```
/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler Transformer Decodable
```

```
/// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.
///
/// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension NormalizationScaler Transformer Sendable where
Element Sendable
```

```
/// An estimator that replaces missing values in the numeric input.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct NumericImputer Element Estimator where
Element BinaryFloatingPoint Element Decodable Element
Encodable

/// The transformer type created by this estimator.
public typealias Transformer ImputeTransformer Element
```

```
/// An imputation strategy.  
public enum Strategy  
  
    /// Imputation strategy that replaces missing elements with the median.  
    case median  
  
    /// Imputation strategy that replaces missing elements with the mean.  
    case mean  
  
    /// Imputation strategy that replaces missing elements with a constant.  
    case constant Element
```

```
/// The imputation strategy.  
public var strategy NumericImputer Element Strategy
```

```
/// Creates an imputer with a strategy.  
public init _ NumericImputer Element Strategy
```

```
/// Creates an imputer with a constant value to use when replacing missing  
values.
```

```
public init Element
```

```
/// Fits a numeric imputer to a sequence of elements.  
///
```

```
/// - Parameters:
```

```
///   - input: A sequence of elements.  
///   - eventHandler: An event handler.
```

```
/// - Returns: The fitted transformer.
```

```
public func fitted S
```

```
EventHandler nil NumericImputer Element Transformer  
where S Sequence S Element Element
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
extension NumericImputer UpdatableEstimator
```

```
/// Creates a default-initialized impute transformer suitable for incremental  
fitting.
```

```
///
```

```
/// - Note: You can't use incremental fitting with an impute transformer  
when using the `median` strategy.
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
```

```
public func makeTransformer
```

```
NumericImputer Element Transformer
```

```
/// Updates an impute transformer with a new sequence of examples.
```

```
///
```

```

    /// - Note: You can't update an impute transformer when using the
`median` strategy.
    /**
     * - Parameters:
     *   - transformer: A transformer to update.
     *   - input: A sequence of examples.
     *   - eventHandler: An event handler.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public func update _ inout
    ImputeTransformer Element some Sequence Element
                    EventHandler nil throws

    /// Encodes the transformer to an encoder.
    /**
     * - Parameters:
     *   - transformer: A transformer this estimator creates.
     *   - encoder: An encoder.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public func encodeWithOptimizer _ inout any
    NumericImputer Element Transformer
                    EstimatorEncoder throws

    /// Reads the encoded transformer with a decoder.
    /**
     * - Parameter decoder: A decoder.
     * - Returns: The decoded transformer.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

    public func decodeWithOptimizer inout any
    EstimatorDecoder throws
    NumericImputer Element Transformer

    /**
     * - Parameters:
     *   - element: The element to impute.
     *   - strategy: The strategy to use.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    extension NumericImputer where Element: Sendable

    /**
     * - Parameters:
     *   - strategy: The strategy to use.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    extension NumericImputerStrategy where Element: Sendable

    /**
     * An object detection annotation.
     */
    /**
     * The annotation consists of a list of bounding boxes and object labels for each
     * image.
     */
    @available(macOS 14.0, iOS 17.0, tvOS 17.0)

```

```

public struct ObjectDetectionAnnotation Label Decodable
Equatable where Label Comparable Label Decodable Label
Encodable Label Hashable

/// The annotation represented by an object label and its bounding box.
public struct Annotation Equatable

/// The bounding box that describes the spatial location of the object.
public let boundingBox CGRect

/// The object label.
public let label Label

/// Coding keys for Annotation.
public enum CodingKeys String CodingKey

case boundingBox

case label

/// Creates a new instance with the specified raw value.
///
/// If there is no value of the type that corresponds with the
specified raw
/// value, this initializer returns `nil`. For example:
///
///     enum PaperSize: String {
///         case A4, A5, Letter, Legal
///     }
///
///     print(PaperSize(rawValue: "Legal"))
///     // Prints "Optional("PaperSize.Legal")"
///
///     print(PaperSize(rawValue: "Tabloid"))
///     // Prints "nil"
///
/// - Parameter rawValue: The raw value to use for the new
instance.

public init String

/// Creates a new instance from the given string.
///
/// If the string passed as `stringValue` does not correspond
to any instance
/// to any instance
/// of this type, the result is `nil`.
///
/// - parameter stringValue: The string value of the
desired key.

public init String

```

```

    /// Creates a new instance from the specified integer.
    ///
    /// If the value passed as `intValue` does not correspond to
any instance of
    /// this type, the result is `nil`.
    ///
    /// - parameter intValue: The integer value of the desired
key.

public init Int

    /// The raw type that can be used to represent all values of the
conforming
    /// type.
    ///
    /// Every distinct value of the conforming type has a
corresponding unique
    /// value of the `RawValue` type, but there may be values of the
`RawValue`
    /// type that don't have a corresponding value of the conforming
type.

@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias RawValue String

    /// The value to use in an integer-indexed collection (e.g. an int-
keyed
    /// dictionary).
public var intValue Int get

    /// The corresponding value of the raw type.
    ///
    /// A new instance initialized with `rawValue` will be equivalent
to this
    /// instance. For example:
    ///
    ///     enum PaperSize: String {
    ///         case A4, A5, Letter, Legal
    ///     }
    ///
    ///     let selectedSize = PaperSize.Letter
    ///     print(selectedSize.rawValue)
    ///     // Prints "Letter"
    ///
    ///     print(selectedSize == PaperSize(rawValue:
selectedSize.rawValue)!)
    ///     // Prints "true"
public var rawValue String get

    /// The string to use in a named collection (e.g. a string-keyed
dictionary).
public var stringValue String get

```

```
    /// Returns a Boolean value indicating whether two values are equal.  
    ///  
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    ///  
    /// - Parameters:  
    ///   - lhs: A value to compare.  
    ///   - rhs: Another value to compare.  
    public static func  
ObjectDetectionAnnotation Label Annotation  
ObjectDetectionAnnotation Label Annotation Bool  
  
    /// The name of the image file.  
    public let imageFileName String  
  
    /// The list of object annotations in the image.  
    public let objects  
ObjectDetectionAnnotation Label Annotation  
  
    /// The most prominent object in the image.  
    public let prominentObject Label  
  
    /// Coding keys for object detection annotations  
    public enum CodingKeys String CodingKey  
  
        case imageFileName  
  
        case objects  
  
        case prominentObject  
  
    /// Creates a new instance with the specified raw value.  
    ///  
    /// If there is no value of the type that corresponds with the specified  
raw  
    /// value, this initializer returns `nil`. For example:  
    ///  
    ///     enum PaperSize: String {  
    ///         case A4, A5, Letter, Legal  
    ///     }  
    ///  
    ///     print(PaperSize(rawValue: "Legal"))  
    ///     // Prints "Optional("PaperSize.Legal")"  
    ///  
    ///     print(PaperSize(rawValue: "Tabloid"))  
    ///     // Prints "nil"
```

```
///  
/// - Parameter rawValue: The raw value to use for the new  
instance.  
public init String  
  
/// Creates a new instance from the given string.  
///  
/// If the string passed as `stringValue` does not correspond to any  
instance  
/// of this type, the result is `nil`.  
///  
/// - parameter stringValue: The string value of the desired  
key.  
public init String  
  
/// Creates a new instance from the specified integer.  
///  
/// If the value passed as `intValue` does not correspond to any  
instance of  
/// this type, the result is `nil`.  
///  
/// - parameter intValue: The integer value of the desired key.  
public init Int  
  
/// The raw type that can be used to represent all values of the  
conforming  
/// type.  
///  
/// Every distinct value of the conforming type has a corresponding  
unique  
/// value of the `RawValue` type, but there may be values of the  
`RawValue`  
/// type that don't have a corresponding value of the conforming type.  
@available iOS 17.0 tvOS 17.0 macOS 14.0  
public typealias RawValue String  
  
/// The value to use in an integer-indexed collection (e.g. an int-keyed  
/// dictionary).  
public var intValue Int get  
  
/// The corresponding value of the raw type.  
///  
/// A new instance initialized with `rawValue` will be equivalent to this  
/// instance. For example:  
///  
///     enum PaperSize: String {  
///         case A4, A5, Letter, Legal  
///     }  
///  
///     let selectedSize = PaperSize.Letter
```

```
    ///      print(selectedSize.rawValue)
    ///      // Prints "Letter"
    ///
    ///      print(selectedSize == PaperSize(rawValue:
selectedSize.rawValue)!)
    ///      // Prints "true"
public var rawValue String get

    /// The string to use in a named collection (e.g. a string-keyed
dictionary).
public var stringValue String get
```

```
    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
ObjectDetectionAnnotation Label
ObjectDetectionAnnotation Label Bool

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Identifiable
```

```
    /// The identifier of the object detection annotation
public var id String get

    /// A type representing the stable identity of the entity associated with
    /// an instance.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias ID String
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Sendable where Label
Sendable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Annotation Decodable
```

```
    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Annotation Sendable
where Label Sendable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation CodingKeys Equatable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation CodingKeys Hashable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation CodingKeys RawRepresentable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Annotation CodingKeys Equatable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Annotation CodingKeys Hashable
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionAnnotation Annotation CodingKeys RawRepresentable
```

```
    /// Metrics for object detection model.
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
```

```
public struct ObjectDetectionMetrics Label where Label
Comparable Label Decodable Label Encodable Label
Hashable

    /// A set of labels present in the dataset.
    public var labels Set Label

    /// The default confidence threshold. It is used as the confidence threshold
    for any label which does not have an explicit confidence threshold, while calculating
        /// averagePrecision and meanAveragePrecision.
    public var defaultConfidenceThreshold Float

    public init

    /// Extracts all the labels from a list of annotations.
    /// - Parameter annotations: A list of annotations.
    /// - Returns: A set of all the labels present in the annotations.
    public static func extractLabels
ObjectDetectionAnnotation Label Set Label

    /// Calculates average precision for all the labels at the bounding box overlap
    threshold.
    ///
    /// - Parameters:
    ///   - predictions: A list of all the predictions from an object detection
    model. Each element in the list is a list of predictions from one image.
    ///   - annotations: A list of all the annotations. Each element is an
    `ObjectDetectionAnnotation` object from one image.
    ///   - confidenceThresholds: Probability thresholds for each label.
    The values will always be between 0.0 and 1.0.
    /// If any label does not have a threshold, the
    `defaultConfidenceThreshold` is used for that label. The default value is
    `[:]`.
    /// - overlapThreshold: The overlap threshold for the bounding
    boxes. The value will always be between 0.0 and 1.0. The default value is `0.5`.
    ///
    /// - Returns: Average precision for all the labels at the overlap
    threshold.
    public func averagePrecision Scalar
DetectedObject Label
ObjectDetectionAnnotation Label
Label Float Double 0.5
Label Scalar where Scalar BinaryFloatingPoint

    /// Calculates the mean average precision at the bounding box overlap
    threshold.
    ///
    /// - Parameters:
    ///   - predictions: A list of all the predictions from an object detection
    model. Each element in the list is a list of predictions from one image.
```

```

    /// - annotations: A list of all the annotations. Each element is an
`ObjectDetectionAnnotation` object from one image.
    /// - confidenceThresholds: Confidence thresholds for each label.
The values will always be between 0.0 and 1.0.
    /// If any label does not have a threshold, the
`defaultConfidenceThreshold` is used for that label. The default value is
`[:]`.
    /// - overlapThreshold: The overlap threshold for the bounding
boxes. The value will always be between 0.0 and 1.0. The default value is `0.5`.
    ///
    /// - Returns: The mean average precision at the overlap threshold.
public func meanAveragePrecision Scalar
    DetectedObject Label
    ObjectDetectionAnnotation Label
    Label Float Double 0.5
    Scalar where Scalar BinaryFloatingPoint

    /// Calculates average of average precision for all the labels, computed at
varied bounding box overlap thresholds.
    /// The overlap thresholds range is from `[0.05, 0.95]` with a stride of
`0.05`.
    ///
    /// - Parameters:
    /// - predictions: A list of all the predictions from an object detection
model. Each element in the list is a list of predictions from one image.
    /// - annotations: A list of all the annotations. Each element is an
`ObjectDetectionAnnotation` object from one image.
    /// - confidenceThresholds: Confidence thresholds for each label.
The values will always be between 0.0 and 1.0.
    /// If any label does not have a threshold, the
`defaultConfidenceThreshold` is used for that label. The default value is
`[:]`.
    ///
    /// - Returns: Average of average precision for all the labels, computed
at varied bounding box overlap thresholds.
public func
averageOfAveragePrecisionAtVariedThresholds Scalar
    DetectedObject Label
    ObjectDetectionAnnotation Label
    Label Float Label Scalar where Scalar
    BinaryFloatingPoint

    /// Calculates the average of mean average precision, computed at varied
bounding box overlap thresholds.
    /// The overlap thresholds range is from `[0.05, 0.95]` with a stride of
`0.05`.
    ///
    /// - Parameters:
    /// - predictions: A list of all the predictions from an object detection
model. Each element in the list is a list of predictions from one image.

```

```

    /**
     - annotations: A list of all the annotations. Each element is an
`ObjectDetectionAnnotation` object from one image.
    /**
     - confidenceThresholds: Confidence thresholds for each label.
The values will always be between 0.0 and 1.0.
    /**
     If any label does not have a threshold, the
`defaultConfidenceThreshold` is used for that label. The default value is
`[:]`.
    /**
    /**
     - Returns: The average of mean average precision, computed at
varied bounding box overlap thresholds.
public func
averageOfMeanAveragePrecisionAtVariedThresholds Scalar
    DetectedObject Label
    ObjectDetectionAnnotation Label
    Label Float           Scalar where Scalar
    BinaryFloatingPoint

```

@available macOS 14.0 iOS 17.0 tvOS 17.0
extension ObjectDetectionMetrics **Sendable where Label**
Sendable

```

    /**
     An estimator that encodes categorical values to an integer array.
    /**
    /**
     The encoded array has an element count equal to the number of categories to
encode.
    /**
     The encoded array for a given category has repeating zero values except at
one index where the value is 1.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct OneHotEncoder Category Estimator where
Category Comparable Category Decodable Category
Encodable Category Hashable

    /**
     Creates a one-hot encoding estimator.
public init

    /**
     Fits a one-hot encoder to a sequence of categories.
    /**
    /**
     - Parameters:
    /**
     - input: A sequence of categories.
    /**
     - eventHandler: An event handler.
    /**
     - Returns: The fitted transformer.
public func fitted S           S
EventHandler nil throws
OneHotEncoder Category Transformer where S Sequence
S Element Category

```

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

```
extension OneHotEncoder : UpdatableEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

        public func makeTransformer
        OneHotEncoder Category Transformer

            /// Updates a transformer with a new sequence of examples.
            ///
            /// - Note: You can't add new categories on subsequent updates. All
            categories should be present in the initial
            /// update.
            ///
            /// - Parameters:
            ///   - transformer: A transformer to update.
            ///   - input: A sequence of examples.
            ///   - eventHandler: An event handler.
            @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

                public func update _ inout
                OneHotEncoder Category Transformer some
                Sequence Category EventHandler nil throws

                    /// Encodes the transformer to an encoder.
                    ///
                    /// - Parameters:
                    ///   - transformer: A transformer this estimator creates.
                    ///   - encoder: An encoder.
                    @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

                        public func encodeWithOptimizer _ inout any
                        OneHotEncoder Category Transformer EstimatorEncoder throws

                            /// Reads the encoded transformer with a decoder.
                            ///
                            /// - Parameter decoder: A decoder.
                            /// - Returns: The decoded transformer.
                            @available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

                                public func decodeWithOptimizer inout any
                                EstimatorDecoder throws
                                OneHotEncoder Category Transformer

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension OneHotEncoder : Sendable where Category : Sendable
```

```
extension OneHotEncoder

    /// A transformer that encodes a category as an array of integers.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public struct Transformer : Transformer

    /// Unique values to encode
public var categories : Set<Category>

    /// Creates a one-hot encoder.
///
/// - Parameter categories: The unique values used for
encoding.
public init categories : Set<Category>

    /// Performs a one-hot encoding on a single input.
///
/// - Parameters
///   - input: A category to encode.
///   - eventHandler: An event handler.
/// - Returns: A one-hot encoded array.
public func applied(_ category : Category) throws Int

    /// Performs a one-hot encoding on a sequence of inputs.
///
/// - Parameters:
///   - input: A sequence of input values.
///   - eventHandler: An event handler.
/// - Returns: An array of one-hot encoded arrays.
public func applied<S : Sequence>(eventHandler : EventHandler<S>) throws [Int] where S.Element == Category

    /// Retrieves the category at the one-hot encoding index.
///
/// - Parameter index: The index of the category.
/// - Returns: The category at the specified index.
public func category(at index : Int) throws Category

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
public typealias Input == Category

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
```

```
13.0
```

```
public typealias Output = Int
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension OneHotEncoder Transformer
CustomDebugStringConvertible
```

```
/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer. This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
///     struct Point: CustomDebugStringConvertible {
///         let x: Int, y: Int
///
///         var debugDescription: String {
///             return "(\(x), \(y))"
///         }
///     }
///
///     let p = Point(x: 21, y: 30)
///     let s = String(reflecting: p)
///     print(s)
///     // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription: String { get }
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension OneHotEncoder Transformer where
Category: Sendable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension OneHotEncoder Transformer: Encodable
```

```
/// Encodes a one-hot encoding transformer.
/// - Parameter encoder: an encoder.
public func encode() throws Encoder
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
extension OneHotEncoder Transformer Decodable

    /// Creates a one-hot encoding transformer from a decoder.
    /// - Parameter decoder: a decoder.
public init any Decoder throws

/// An optimization error.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public enum OptimizationError : LocalizedError : Equatable
Sendable

    /// Numerical underflow (not enough progress).
case numericUnderflow

    /// Numerical overflow. Step size parameter may be too large.
case numericOverflow

    /// The optimization operation is not supported on this platform.
case unsupportedPlatform

    /// A localized message describing what error occurred.
public var errorDescription : String get

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
///
/// Implement this method to conform to the `Hashable` protocol. The
/// components used for hashing must be the same as the components
compared
/// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
/// with each of these components.
///
/// - Important: In your implementation of `hash(into:)`,
/// don't call `finalize()` on the `hasher` instance provided,
/// or replace it with a different instance.
/// Doing so may become a compile-time error in the future.
///
/// - Parameter hasher: The hasher to use when combining the
components
/// of this instance.
public func hash inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
///
/// Equality is the inverse of inequality. For any values `a` and `b`,
/// `a == b` implies that `a != b` is `false`.
///
/// - Parameters:
```

```
    /// - lhs: A value to compare.
    /// - rhs: Another value to compare.
    public static func      OptimizationError
OptimizationError      Bool

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
        /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
    public var hashValue  Int   get

@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0
extension OptimizationError  CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription  String   get

@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0
extension OptimizationError  Hashable

    /// A linear optimization strategy.
@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0
public enum OptimizationStrategy  Hashable  Codable  Sendable

    /// Chooses the best optimization strategy based on the problem size and
configuration.
    case automatic

    /// An optimization strategy that can handle non-smooth problems.
    ///
    /// Select this strategy when using L1 regularization. Using L1 regularization
causes the optimization problem to
        /// be non-smooth. Other optimization strategies rely on the optimization
problem being smooth and will likely fail
    /// to converge when using L1 regularization.
    case nonSmooth

    /// An optimization strategy that minimizes memory use.
    ///
    /// Select this strategy when solving large problems that run out of memory
```

```
when using the `fast` optimization
```

```
/// strategy, or when you need to keep memory use low.
```

```
case lowMemory
```

```
/// An optimization strategy that minimizes computation time.
```

```
///
```

```
/// Select this strategy for smooth problems that fit in memory.
```

```
case fast
```

```
/// Returns a Boolean value indicating whether two values are equal.
```

```
///
```

```
/// Equality is the inverse of inequality. For any values `a` and `b` ,
```

```
/// `a == b` implies that `a != b` is `false` .
```

```
///
```

```
/// - Parameters:
```

```
///   - lhs: A value to compare.
```

```
///   - rhs: Another value to compare.
```

```
public static func OptimizationStrategy
```

```
OptimizationStrategy Bool
```

```
/// Hashes the essential components of this value by feeding them into the  
/// given hasher.
```

```
///
```

```
/// Implement this method to conform to the `Hashable` protocol. The  
/// components used for hashing must be the same as the components  
compared
```

```
/// in your type's `==` operator implementation. Call
```

```
`hasher.combine(_:)`
```

```
/// with each of these components.
```

```
///
```

```
/// - Important: In your implementation of `hash(into:)` ,  
/// don't call `finalize()` on the `hasher` instance provided,  
/// or replace it with a different instance.
```

```
/// Doing so may become a compile-time error in the future.
```

```
///
```

```
/// - Parameter hasher: The hasher to use when combining the  
components
```

```
/// of this instance.
```

```
public func hash inout Hasher
```

```
/// Encodes this value into the given encoder.
```

```
///
```

```
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.
```

```
///
```

```
/// This function throws an error if any values are invalid for the given  
/// encoder's format.
```

```
///
```

```
/// - Parameter encoder: The encoder to write data to.
```

```
public func encode any Encoder throws
```

```

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable` requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

/// A transformer that unwraps optional elements and throws when encountering
missing values.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct OptionalUnwrapper Element Transformer

    /// Creates a transformer that unwraps an optional element or throws if the
value is nil.
public init

    /// Unwraps an optional element or throws if the value is `nil`.
    ///
    /// - Parameters:
    ///   - input: The optional input.
    ///   - eventHandler: An event handler.
    /// - Returns: The unwrapped value.
    /// - Throws: `MissingValueError` if the input is `nil`.
@inlinable public func applied Element
    EventHandler nil throws Element

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Input Element

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Output Element

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension OptionalUnwrapper Sendable where Element
Sendable

/// An ordinal encoder estimator encodes categorical values to ordinal integer
values.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct OrdinalEncoder<Category : Comparable, Category : Decodable, Category : Encodable, Category : Hashable> : Estimator<Category>

/// Creates an ordinal encoding estimator.
public init

/// Fits an ordinal encoder to a sequence of categories.
///
/// - Parameters:
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
/// - Returns: The fitted transformer.
public func fitted<S : Sequence, S : Element, S : Category>(
    EventHandler nil throws
) -> OrdinalEncoder<Category> Transformer where S.Element == Category

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension OrdinalEncoder : UpdatableEstimator

/// Creates a default-initialized transformer suitable for incremental fitting.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public func makeTransformer() -> OrdinalEncoder<Category> Transformer

/// Updates a transformer with a new sequence of examples.
///
/// - Note: You can't add new categories on subsequent updates. All
/// categories should be present in the initial
/// update.
///
/// - Parameters:
///   - transformer: A transformer to update.
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

```

```

public func update _ inout OrdinalEncoder Category Transformer some  

Sequence Category EventHandler nil throws  

/// Encodes the transformer to an encoder.  

///  

/// - Parameters:  

/// - transformer: A transformer this estimator creates.  

/// - encoder: An encoder.  

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  

  

public func encodeWithOptimizer _ inout any  

OrdinalEncoder Category Transformer throws  

EstimatorEncoder throws  

/// Reads the encoded transformer with a decoder.  

///  

/// - Parameter decoder: A decoder.  

/// - Returns: The decoded transformer.  

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  

  

public func decodeWithOptimizer inout any  

EstimatorDecoder throws  

OrdinalEncoder Category Transformer  

  

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  

extension OrdinalEncoder Sendable where Category Sendable

```

extension **OrdinalEncoder**

```

/// A transformer that encodes a category as an integer.  

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

public struct **Transformer** **Transformer**

```

/// Unique values to encode

```

```

public var categories Set Category

```

```

/// Creates an ordinal encoder.

```

```

///

```

```

/// - Parameter categories: The unique values used for  

encoding.

```

```

public init Set Category

```

```

/// Performs an ordinal encoding on a single input.

```

```

///

```

```

    /// - Parameters:
    ///   - input: A category to encode.
    ///   - eventHandler: An event handler.
    /// - Returns: A ordinal encoded value.
    public func applied Category
EventHandler nil throws Int

    /// Performs an ordinal encoding on a sequence of inputs.
    ///

    /// - Parameters:
    ///   - input: A sequence of input values.
    ///   - eventHandler: An event handler.
    /// - Returns: An array of ordinal encoded values.
    public func applied S _ S
EventHandler nil throws [Int] where S : Sequence
S.Element == Category

    /// Retrieves the category at the ordinal encoding index.
    ///

    /// - Parameter index: The index of the category.
    /// - Returns: The category at the specified index.
    public func category Int Category

    /// The input type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
    public typealias Input Category

    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0
    public typealias Output Int

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension OrdinalEncoder Transformer
CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///

    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)` initializer.
    /// This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///

    /// struct Point: CustomDebugStringConvertible {
    ///   let x: Int, y: Int

```

```
///  
///     var debugDescription: String {  
///         return "(\(x), \(y))"  
///     }  
/// }  
  
/// let p = Point(x: 21, y: 30)  
/// let s = String(reflecting: p)  
/// print(s)  
/// // Prints "(21, 30)"  
///  
/// The conversion of `p` to a string in the assignment to `s` uses the  
/// `Point` type's `debugDescription` property.  
public var debugDescription String get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension OrdinalEncoder Transformer Sendable where Category  
Sendable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension OrdinalEncoder Transformer Encodable
```

```
/// Encodes an ordinal encoding transformer.  
/// – Parameter encoder: an encoder.  
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension OrdinalEncoder Transformer Decodable
```

```
/// Creates an ordinal encoding transformer from a decoder.  
/// – Parameter decoder: a decoder.  
public init any Decoder throws
```

```
/// Errors related to pipeline data affinity problems.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public enum PipelineDataError LocalizedError Equatable  
Sendable
```

```
/// An error that indicates that an expected value is missing.  
case missingValue String
```

```
/// An error that indicates that an expected annotation is missing.  
case missingAnnotation String
```

```
/// An error that indicates that the input to fit is empty.
```

```
case emptyInput           String
    /// An error that indicates that a new category was encountered after fitting.
    case unrecognizedCategory   String
String

    /// An error that indicates that an input doesn't have the expected data
format.
    case incompatibleDataFormat     String
        String

    /// An error that indicates that an input is not compatible with an operation's
configuration.
    case incompatibleConfiguration  String
        String

    /// An error that indicates that an input's doesn't have the expected shape for
the operation.
    case incompatibleShape    Int           String
    /// A localized message describing what error occurred.
public var errorDescription  String      get
    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func      PipelineDataError
PipelineDataError      Bool

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension PipelineDataError  CustomDebugStringConvertible

    /// A text representation of the error.
public var debugDescription  String      get
    /// A pose that contains joint keypoints from a person, a hand, or a combination.
@available macOS 13.0  iOS 16.0  tvOS 16.0
public struct Pose   Sendable
    /// Creates a pose from a body or hand pose observation.
    ///
    /// - Parameters:
    ///   - observation: Recognized points observation that comes from
```

either a body pose or hand pose request.

```
public init _ VNRecognizedPointsObservation
throws

    /// Creates a pose from a dictionary of joint keypoints.
    ///
    /// - Parameters:
    ///   - points: A dictionary of pose joint keypoints, where keys are joint
    names and values are joint points.
    public init JointKey JointPoint

    /// A dictionary of all keypoints in the pose
    public var keypoints JointKey JointPoint

    /// Computes the bounding box area of the pose.
    ///
    /// - Parameters:
    ///   - confidenceThreshold: a threshold confidence between 0 to 1
    for the keypoints to be considered valid to compute the bounding box area. The
    default value is 0.2.
    public func boundingBoxArea Float
0.2 Float
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension Pose Encodable
```

```
    /// Encodes the pose value into the given encoder.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public func encode any Encoder throws
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension Pose Decodable
```

```
    /// Creates a pose by decoding from the given decoder.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public init any Decoder throws
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0
extension Pose Equatable
```

```
    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
```

```
    /// - rhs: Another value to compare.
public static func      Pose      Pose      Bool

    /// Pose selection strategy.
@available macOS 13.0  iOS 16.0  tvOS 16.0
public enum PoseSelectionStrategy  Sendable

    /// The strategy to choose a pose with the maximum bounding box area.
    case maximumBoundingBoxArea

    /// The strategy to choose a pose where a joint in it has the highest y
    coordinate location.
    case highestJointLocation

    /// The strategy to choose a pose where a joint in it has the lowest y
    coordinate location.
    case lowestJointLocation

    /// The strategy to choose a pose where a joint in it has the leftmost x
    coordinate location.
    case leftmostJointLocation

    /// The strategy to choose a pose where a joint in it has the rightmost x
    coordinate location.
    case rightmostJointLocation

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func      PoseSelectionStrategy
PoseSelectionStrategy      Bool

    /// Hashes the essential components of this value by feeding them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable` protocol. The
    /// components used for hashing must be the same as the components
    compared
        /// in your type's `==` operator implementation. Call
        `hasher.combine(_:)`
        /// with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`,
        /// don't call `finalize()` on the `hasher` instance provided,
```

```

    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable` requirement. To
conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension PoseSelectionStrategy Equatable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0
extension PoseSelectionStrategy Hashable
```

```

    /// A transformer that selects one pose from an array of poses.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public struct PoseSelector Transformer Sendable

    /// Pose selection strategy.
public var strategy PoseSelectionStrategy

    /// A threshold confidence between 0 to 1 for the joints to be considered valid
in pose selection. The default value is 0.2.
public var confidenceThreshold Float

    /// Creates a pose selector.
    /// - Parameters:
    /// - strategy: The strategy used to choose a pose if multiple poses
are detected on the same frame. Default strategy is to select a pose with maximum
bounding box area.
    /// - confidenceThreshold: A threshold confidence between 0 to 1
for the joints to be considered valid in pose selection. The default value is 0.2.
public init PoseSelectionStrategy
                    Float

```

```
    /// Creates a pose selector.  
    /// - Parameters:  
    ///   - strategy: The strategy used to choose a pose if multiple poses  
    /// are detected on the same frame. Default strategy is to select a pose with maximum  
    /// bounding box area.  
    public init PoseSelectionStrategy  
  
    /// Creates a pose selector.  
    public init  
  
    /// Select a pose if multiple poses are detected on the same frame.  
    ///  
    /// - Parameters:  
    ///   - input: An array of poses.  
    ///   - eventHandler: An event handler.  
    /// - Returns: A selected pose based on the strategy.  
    public func applied Pose  
EventHandler nil Pose  
  
    /// The input type.  
@available iOS 16.0 tvOS 16.0 macOS 13.0  
public typealias Input Pose  
  
    /// The output type.  
@available iOS 16.0 tvOS 16.0 macOS 13.0  
public typealias Output Pose  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
extension PoseSelector CustomDebugStringConvertible  
  
    /// A textual representation of this instance, suitable for debugging.  
    ///  
    /// Calling this property directly is discouraged. Instead, convert an  
    /// instance of any type to a string by using the `String(reflecting:)`  
    /// initializer. This initializer works with any type, and uses the custom  
    /// `debugDescription` property for types that conform to  
    /// `CustomDebugStringConvertible`:  
    ///  
    ///     struct Point: CustomDebugStringConvertible {  
    ///         let x: Int, y: Int  
    ///  
    ///         var debugDescription: String {  
    ///             return "(\(x), \(y))"  
    ///         }  
    ///     }  
    ///  
    ///     let p = Point(x: 21, y: 30)  
    ///     let s = String(reflecting: p)  
    ///     print(s)
```

```

    /// Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

    /// An asynchronous sequence of eagerly stored temporal features.
    ///
    /// This sequence eagerly stores the results of preprocessing a temporal
    estimator.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public struct PreprocessedFeatureSequence Feature
    TemporalSequence

        /// The stored temporal features.
        public var features TemporalFeature Feature

        /// The stored temporal feature count.
        @inlinable public var count Int get

        /// Creates an asynchronous sequence of stored temporal features.
        ///
        /// - Parameter sequence: An asynchronous sequence of temporal
        features.
        public init S _ S async throws where Feature
        S Feature S TemporalSequence

        /// Creates the asynchronous iterator that produces elements of this
        /// asynchronous sequence.
        ///
        /// - Returns: An instance of the `AsyncIterator` type used to
        produce
        /// elements of the asynchronous sequence.
        public func makeAsyncIterator
        PreprocessedFeatureSequence Feature AsyncIterator

        /// The type of element produced by this asynchronous sequence.
        @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

        public typealias Element TemporalFeature Feature

extension PreprocessedFeatureSequence

    /// The type of asynchronous iterator that produces elements of this
    /// asynchronous sequence.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public struct AsyncIterator AsyncIteratorProtocol

```

```
    /// Asynchronously advances to the next element and returns it, or
ends the
    /// sequence if there is no next element.
    ///
    /// - Returns: The next element, if it exists, or `nil` to signal the
end of
    /// the sequence.
@inlinable public mutating func next
TemporalFeature Feature

@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
public typealias Element TemporalFeature Feature
```

```
/// An estimator that composes a preprocessing transformer and an estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct PreprocessingEstimator Preprocessor
Estimator Estimator where Preprocessor Transformer
Estimator Estimator Preprocessor Output
Estimator Transformer Input

/// The transformer type created by this estimator.
public typealias Transformer
ComposedTransformer Preprocessor Estimator Transformer

/// The input type.
public typealias Input Preprocessor Input

/// The intermediate type.
public typealias Intermediate Preprocessor Output

/// The output type.
public typealias Output Estimator Transformer Output

/// The preprocessing transformer.
public var preprocessor Preprocessor

/// The estimator.
public var estimator Estimator

/// Creates a composed estimator from a preprocessing transformer and an
estimator.
public init _ Preprocessor _
Estimator

/// Preprocesses a sequence of examples.
///
```

```

    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The preprocessed examples.
    @inlinable public func preprocesses S           S
                           EventHandler nil async throws
Preprocessor Output where S Sequence Preprocessor Input
S Element

    /// Fits a transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted S
                           S           EventHandler nil async
throws PreprocessingEstimator Preprocessor
Estimator Transformer where S Sequence Preprocessor Output
S Element S Element Estimator Transformer Input

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted S           S
                           EventHandler nil async throws
PreprocessingEstimator Preprocessor Estimator Transformer
where S Sequence Preprocessor Input S Element

    /// Encodes a fitted transformer.
    public func encode
PreprocessingEstimator Preprocessor Estimator Transformer
      inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws
PreprocessingEstimator Preprocessor Estimator Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension PreprocessingEstimator Sendable where Preprocessor
Sendable Estimator Sendable

    /// A supervised estimator that composes a preprocessing transformer and a

```

```
supervised estimator.

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct PreprocessingSupervisedEstimator Preprocessor
Estimator SupervisedEstimator where Preprocessor
Transformer Estimator SupervisedEstimator
Preprocessor Output Estimator Transformer Input

    /// The transformer type created by this estimator.
    public typealias Transformer
ComposedTransformer Preprocessor Estimator Transformer

    /// The input type.
    public typealias Input Preprocessor Input

    /// The intermediate type.
    public typealias Intermediate Preprocessor Output

    /// The output type.
    public typealias Output Estimator Transformer Output

    /// The annotation type.
    public typealias Annotation Estimator Annotation

    /// The preprocessing transformer.
    public var preprocessor Preprocessor

    /// The estimator.
    public var estimator Estimator

    /// Creates a composed supervised estimator from a preprocessing
transformer and an estimator.
    public init _ Preprocessor _
Estimator

    /// Prepares a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The prepared examples.
    @inlinable public func prepared S S
EventHandler nil async throws
AnySequence AnnotatedFeature Preprocessor Output
PreprocessingSupervisedEstimator Preprocessor
Estimator Annotation where S Sequence S Element
AnnotatedFeature Preprocessor Input Estimator Annotation

    /// Fits a transformer to a sequence of prepared features.
    ///
    /// - Parameters:
```

```

    /// - preprocessed: A sequence of preprocessed features.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted S
    S           EventHandler   nil async
throws PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer where S Sequence S Element
AnnotatedFeature Preprocessor Output Estimator Annotation

    /// Fits a composed transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessedInput: A sequence of preprocessed features.
    ///   - preprocessedValidation: A sequence of preprocessed
features used for validating the fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted Input
Validation           Input
                           Validation
EventHandler   nil async throws
PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer where Input Sequence Validation
Sequence Input Element
AnnotatedFeature Preprocessor Output Estimator Annotation
Validation Element AnnotatedFeature Preprocessor Output
Estimator Annotation

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted InputSequence
InputSequence           EventHandler   nil async throws
PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
InputSequence Element AnnotatedFeature Preprocessor Input
Estimator Annotation

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted InputSequence

```

```
Validation           InputSequence
Validation           EventHandler   nil  async throws
PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
Validation Sequence InputSequence Element
AnnotatedFeature Preprocessor Input Estimator Annotation
Validation Element AnnotatedFeature Preprocessor Input
Estimator Annotation

    /// Encodes a fitted transformer.
    public func encode _  

PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer           inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws
PreprocessingSupervisedEstimator Preprocessor
Estimator Transformer

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension PreprocessingSupervisedEstimator Sendable where
Preprocessor Sendable Estimator Sendable

/// A supervised tabular estimator that composes a preprocessing transformer and
a supervised tabular estimator.
@available macOS 14.0  iOS 17.0  tvOS 17.0  watchOS 11.0
public struct
PreprocessingSupervisedTabularEstimator Preprocessor
Estimator SupervisedTabularEstimator where Preprocessor
TabularTransformer Estimator SupervisedTabularEstimator

    /// The transformer type created by this estimator.
    public typealias Transformer
ComposedTabularTransformer Preprocessor
Estimator Transformer

    /// The input type.
    public typealias Input     Preprocessor Input

    /// The intermediate type.
    public typealias Intermediate Preprocessor Output

    /// The output type.
    public typealias Output    Estimator Transformer Output
```

```
    ///> The annotation type.
public typealias Annotation  Estimator Annotation

    ///> The preprocessing transformer.
public var preprocessor  Preprocessor

    ///> The estimator.
public var estimator  Estimator

    ///> The annotation column identifier.
public var annotationColumnID
ColumnID Estimator Annotation

    ///> Creates a composed supervised tabular estimator from a preprocessing
transformer and a supervised tabular estimator.
public init _  Preprocessor _  
Estimator

    ///> Preprocesses a data frame of examples.
    ///
    ///> - Parameters:
    ///>   - input: A data frame of examples.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The preprocessed examples.
@inlinable public func preprocessed
    EventHandler  nil  async throws
    DataFrame  DataFrame

    ///> Fits a transformer to a data frame of preprocessed examples while
validating.
    ///
    ///> - Parameters:
    ///>   - preprocessedInput: A data frame of preprocessed features
used for fitting the transformer.
    ///>   - preprocessedValidation: A data frame of preprocessed
features used for validating the fitted transformer.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
@inlinable public func fitted
    DataFrame
    DataFrame
EventHandler  nil  async throws
PreprocessingSupervisedTabularEstimator  Preprocessor
Estimator  Transformer

    ///> Fits a transformer to a data frame
    ///
    ///> - Parameters:
    ///>   - input: A data frame containing examples used for fitting the
transformer.
    ///>   - validation: A data frame containing examples used for
```

```

validating the fitted transformer.

    /// - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted DataFrame
                                DataFrame
nil async throws DataFrame
PreprocessingSupervisedTabularEstimator Preprocessor
Estimator Transformer

    /// Encodes a fitted transformer.
public func encode _ DataFrame
PreprocessingSupervisedTabularEstimator Preprocessor
Estimator Transformer inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
public func decode inout any
EstimatorDecoder throws
PreprocessingSupervisedTabularEstimator Preprocessor
Estimator Transformer

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension PreprocessingSupervisedTabularEstimator Sendable
where Preprocessor Sendable Estimator Sendable

/// A supervised temporal estimator that composes a preprocessing transformer
and a supervised temporal estimator.

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public struct
PreprocessingSupervisedTemporalEstimator Preprocessor
Estimator SupervisedTemporalEstimator where Preprocessor
TemporalTransformer Estimator SupervisedTemporalEstimator
Preprocessor Output Estimator Transformer Input

    /// The transformer type created by this estimator.
public typealias Transformer
ComposedTemporalTransformer Preprocessor
Estimator Transformer

    /// The input type.
public typealias Input Preprocessor Input

    /// The intermediate type.
public typealias Intermediate Preprocessor Output

```

```

    ///> The output type.
public typealias Output    Estimator Transformer Output

    ///> The annotation type.
public typealias Annotation    Estimator Annotation

    ///> The preprocessing transformer.
public var preprocessor  Preprocessor

    ///> The estimator.
public var estimator  Estimator

    ///> Creates a composed supervised temporal estimator from a preprocessing
transformer and a supervised temporal estimator.
public init _           Preprocessor _  

Estimator

    ///> Preprocesses a sequence of examples.
    ///>
    ///> - Parameters:
    ///>   - input: A sequence of examples.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The preprocessed examples.
@inlinable public func preprocessed InputSequence  

FeatureSequence          InputSequence  

EventHandler  nil async throws  

    AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out  

put  PreprocessingSupervisedTemporalEstimator Preprocessor  

Estimator Annotation where InputSequence Sequence  

FeatureSequence TemporalSequence Preprocessor Input  

FeatureSequence Feature InputSequence Element  

AnnotatedFeature FeatureSequence Estimator Annotation

    ///> Fits a transformer to a sequence of preprocessed annotated features.
    ///>
    ///> - Parameters:
    ///>   - preprocessed: A sequence of preprocessed features.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
@inlinable public func fitted  

    AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out  

put  PreprocessingSupervisedTemporalEstimator Preprocessor  

Estimator Annotation           EventHandler nil  

async throws  

PreprocessingSupervisedTemporalEstimator Preprocessor  

Estimator Transformer

    ///> Fits a transformer to a sequence of preprocessed examples while
validating.

```

```

    /**
     * - Parameters:
     *   - preprocessedInput: A sequence of preprocessed features used for fitting the transformer.
     *   - preprocessedValidation: A sequence of preprocessed features used for validating the fitted transformer.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     */
    @inlinable public func fitted

        AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out
        put    PreprocessingSupervisedTemporalEstimator Preprocessor
        Estimator Annotation
        AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out
        put    PreprocessingSupervisedTemporalEstimator Preprocessor
        Estimator Annotation           EventHandler nil
async throws
        PreprocessingSupervisedTemporalEstimator Preprocessor
        Estimator Transformer

    /**
     * Fits a transformer to a sequence of examples.
     */
    /**
     * - Parameters:
     *   - input: A sequence of examples used for fitting the transformer.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     */
    @inlinable public func fitted InputSequence
        FeatureSequence          InputSequence
        EventHandler nil async throws
        PreprocessingSupervisedTemporalEstimator Preprocessor
        Estimator Transformer where InputSequence Sequence
        FeatureSequence TemporalSequence Preprocessor Input
        FeatureSequence Feature InputSequence Element
        AnnotatedFeature FeatureSequence Estimator Annotation

    /**
     * Fits a transformer to a sequence of examples while validating with a validation sequence.
     */
    /**
     * - Parameters:
     *   - input: A sequence of examples used for fitting the transformer.
     *   - validation: A sequence of examples used for validating the fitted transformer.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     */
    @inlinable public func fitted InputSequence Validation
        FeatureSequence          InputSequence
        Validation               EventHandler nil
async throws
        PreprocessingSupervisedTemporalEstimator Preprocessor
        Estimator Transformer where InputSequence Sequence

```

```
Validation Sequence FeatureSequence TemporalSequence
Preprocessor Input FeatureSequence Feature
InputSequence Element AnnotatedFeature FeatureSequence
Estimator Annotation Validation Element
AnnotatedFeature FeatureSequence Estimator Annotation

    /// Encodes a fitted transformer.
    public func encode _  

    PreprocessingSupervisedTemporalEstimator Preprocessor  

    Estimator Transformer inout any  

    EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode inout any  

    EstimatorDecoder throws
    PreprocessingSupervisedTemporalEstimator Preprocessor  

    Estimator Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension PreprocessingSupervisedTemporalEstimator Sendable
where Preprocessor Sendable Estimator Sendable

/// An estimator that composes a preprocessing transformer and an estimator.
available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct PreprocessingTabularEstimator Preprocessor
Estimator TabularEstimator where Preprocessor
TabularTransformer Estimator TabularEstimator

    /// The transformer type created by this estimator.
    public typealias Transformer
    ComposedTabularTransformer Preprocessor
Estimator Transformer

    /// The input type.
    public typealias Input Preprocessor Input

    /// The intermediate type.
    public typealias Intermediate Preprocessor Output

    /// The output type.
    public typealias Output Estimator Transformer Output

    /// The preprocessing transformer.
```

```
public var preprocessor Preprocessor
    /// The estimator.
public var estimator Estimator
    /// Creates a composed estimator from a preprocessing transformer and an
estimator.
public init _ Preprocessor -
Estimator
    /// Prepares a data frame of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The preprocessed examples.
@inlinable public func preprocessed DataFrame
    EventHandler nil async throws DataFrame
    /// Fits a transformer to a data frame of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A data frame of preprocessed features.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted DataFrame
    EventHandler nil async throws Transformer
    PreprocessingTabularEstimator Preprocessor
    Estimator Transformer
    /// Fits a composed transformer to a data frame of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted DataFrame
    EventHandler nil async throws Transformer
    PreprocessingTabularEstimator Preprocessor
    Estimator Transformer
    /// Encodes a fitted transformer.
public func encode inout any
    PreprocessingTabularEstimator Preprocessor
    Estimator Transformer inout any
    EstimatorEncoder throws
    /// Decodes a previously fitted transformer.
public func decode inout any
```

```
EstimatorDecoder throws
PreprocessingTabularEstimator Preprocessor
Estimator Transformer

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension PreprocessingTabularEstimator Sendable where
Preprocessor Sendable Estimator Sendable

/// A temporal estimator that composes a preprocessing transformer and a
temporal estimator.
available 13.0 15.0
available 16.0 18.0
available 16.0 18.0
available 1.0 2.0
available
public struct PreprocessingTemporalEstimator Preprocessor
Estimator TemporalEstimator where Preprocessor
TemporalTransformer Estimator TemporalEstimator
Preprocessor Output Estimator Transformer Input

/// The transformer type created by this estimator.
public typealias Transformer
ComposedTemporalTransformer Preprocessor
Estimator Transformer

/// The input type.
public typealias Input Preprocessor Input

/// The intermediate type.
public typealias Intermediate Preprocessor Output

/// The output type.
public typealias Output Estimator Transformer Output

/// The preprocessing transformer.
public var preprocessor Preprocessor

/// The estimator.
public var estimator Estimator

/// Creates a composed temporal estimator from a preprocessing
transformer and a temporal estimator.
public init _ Preprocessor _
Estimator

/// Preprocesses a sequence of examples.
///
/// - Parameters:
```

```

    /// - input: A sequence of examples.
    /// - eventHandler: An event handler.
    /// - Returns: The preprocessed examples.
    @inlinable public func preprocess InputSequence
        InputSequence           EventHandler   nil async
throws      PreprocessedFeatureSequence Preprocessor Output
where InputSequence Sequence Preprocessor Input
InputSequence Element Feature InputSequence Element
TemporalSequence

    /// Fits a transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A sequence of preprocessed featuress.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted
        PreprocessedFeatureSequence Preprocessor Output
                           EventHandler   nil async throws
PreprocessingTemporalEstimator Preprocessor
Estimator Transformer

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted InputSequence
        InputSequence           EventHandler   nil async throws
PreprocessingTemporalEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
Preprocessor Input InputSequence Element Feature
InputSequence Element TemporalSequence

    /// Encodes a fitted transformer.
    public func encode -
PreprocessingTemporalEstimator Preprocessor
Estimator Transformer           inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws
PreprocessingTemporalEstimator Preprocessor
Estimator Transformer

```

@available

13.0

15.0

```
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension PreprocessingTemporalEstimator Sendable where
Preprocessor Sendable Estimator Sendable

/// An updatable estimator that composes a preprocessing transformer and an
updatable estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct PreprocessingUpdatableEstimator Preprocessor
Estimator UpdatableEstimator where Preprocessor
Transformer Estimator UpdatableEstimator
Preprocessor Output Estimator Transformer Input

/// The transformer type created by this estimator.
public typealias Transformer
ComposedTransformer Preprocessor Estimator Transformer

/// The input type.
public typealias Input Preprocessor Input

/// The intermediate type.
public typealias Intermediate Preprocessor Output

/// The output type.
public typealias Output Estimator Transformer Output

/// The preprocessing transformer.
public var preprocessor Preprocessor

/// The estimator.
public var estimator Estimator

/// Creates a composed updatable estimator from a preprocessing
transformer and an estimator.
public init _ Preprocessor _ Estimator

/// Creates a default-initialized transformer suitable for incremental fitting.
@inlinable public func makeTransformer
PreprocessingUpdatableEstimator Preprocessor
Estimator Transformer

/// Prepares a sequence of examples.
///
/// - Parameters:
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
/// - Returns: The prepared examples.
```

```
@inlinable public func preprocessed S S
    EventHandler nil async throws
Preprocessor Output where S Sequence Preprocessor Input
S Element

    /// Fits a transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted S
    S EventHandler nil async
throws PreprocessingUpdatableEstimator Preprocessor
Estimator Transformer where S Sequence Preprocessor Output
S Element S Element Estimator Transformer Input

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted S S
    S EventHandler nil async throws
PreprocessingUpdatableEstimator Preprocessor
Estimator Transformer where S Sequence Preprocessor Input
S Element

    /// Updates a transformer with a new sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
@inlinable public func update InputSequence _
    inout
PreprocessingUpdatableEstimator Preprocessor
Estimator Transformer
InputSequence EventHandler nil async throws
where InputSequence Sequence Preprocessor Output
InputSequence Element InputSequence Element
Estimator Transformer Input

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
```

```

    /// - eventHandler: An event handler.
    @inlinable public func update InputSequence _  

        inout  

    PreprocessingUpdatableEstimator Preprocessor  

    Estimator Transformer InputSequence  

        EventHandler nil async throws where  

    InputSequence Sequence Preprocessor Input  

    InputSequence Element

    /// Encodes a fitted transformer.  

    public func encode _  

    PreprocessingUpdatableEstimator Preprocessor  

    Estimator Transformer inout any  

    EstimatorEncoder throws

    /// Decodes a previously fitted transformer.  

    public func decode inout any  

    EstimatorDecoder throws  

    PreprocessingUpdatableEstimator Preprocessor  

    Estimator Transformer

    /// Encodes the transformer and optimizer to an encoder.  

    /// - Parameters:  

    ///   - transformer: A transformer this estimator creates.  

    ///   - encoder: An encoder.  

    public func encodeWithOptimizer _  

    PreprocessingUpdatableEstimator Preprocessor  

    Estimator Transformer inout any  

    EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer.  

    /// - Parameter decoder: A decoder.  

    /// - Returns: The decoded transformer.  

    public func decodeWithOptimizer inout any  

    EstimatorDecoder throws  

    PreprocessingUpdatableEstimator Preprocessor  

    Estimator Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension PreprocessingUpdatableEstimator Sendable where
Preprocessor Sendable Estimator Sendable

/// An updatable supervised estimator that composes a preprocessing transformer
and an updatable supervised estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

```
public struct  
PreprocessingUpdatableSupervisedEstimator Preprocessor  
Estimator UpdatableSupervisedEstimator where Preprocessor  
Transformer Estimator UpdatableSupervisedEstimator  
Preprocessor Output Estimator Transformer Input  
  
    /// The transformer type created by this estimator.  
    public typealias Transformer  
ComposedTransformer Preprocessor Estimator Transformer  
  
    /// The input type.  
    public typealias Input Preprocessor Input  
  
    /// The intermediate type.  
    public typealias Intermediate Preprocessor Output  
  
    /// The output type.  
    public typealias Output Estimator Transformer Output  
  
    /// The annotation type.  
    public typealias Annotation Estimator Annotation  
  
    /// The preprocessing transformer.  
    public var preprocessor Preprocessor  
  
    /// The estimator.  
    public var estimator Estimator  
  
    /// Creates a composed supervised estimator from a preprocessing  
    transformer and a supervised estimator.  
    public init _ Preprocessor _  
Estimator  
  
    /// Creates a default-initialized transformer suitable for incremental fitting.  
    @inlinable public func makeTransformer  
PreprocessingUpdatableSupervisedEstimator Preprocessor  
Estimator Transformer  
  
    /// Prepares a sequence of examples.  
    ///  
    /// - Parameters:  
    ///   - input: A sequence of examples.  
    ///   - eventHandler: An event handler.  
    /// - Returns: The preprocessed examples.  
    @inlinable public func preprocessed S S  
        EventHandler nil async throws  
AnySequence AnnotatedFeature Preprocessor Output  
PreprocessingUpdatableSupervisedEstimator Preprocessor  
Estimator Annotation where S Sequence S Element
```

```

AnnotatedFeature Preprocessor Input Estimator Annotation

    /// Fits a transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted S EventHandler nil async
throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer where S Sequence S Element
AnnotatedFeature Preprocessor Output Estimator Annotation

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - preprocessedInput: A sequence of preprocessed features used for fitting the transformer.
    ///   - preprocessedValidation: A sequence of preprocessed features used for validating the fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted InputSequence
Validation InputSequence
Validation
EventHandler nil async throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
Validation Sequence InputSequence Element
AnnotatedFeature Preprocessor Output Estimator Annotation
Validation Element AnnotatedFeature Preprocessor Output
Estimator Annotation

    /// Fits a composed transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted InputSequence
InputSequence EventHandler nil async throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
InputSequence Element AnnotatedFeature Preprocessor Input
Estimator Annotation

    /// Fits a composed transformer to a sequence of examples.
    ///

```

```

    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
    fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted InputSequence
Validation          InputSequence
Validation           EventHandler  nil  async throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator  Transformer where InputSequence Sequence
Validation  Sequence InputSequence Element
AnnotatedFeature Preprocessor Input Estimator Annotation
Validation Element  AnnotatedFeature Preprocessor Input
Estimator Annotation

    /// Updates a transformer with a new sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    @inlinable public func update InputSequence inout
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator  Transformer
InputSequence           EventHandler  nil  async throws
where InputSequence Sequence InputSequence Element
AnnotatedFeature Preprocessor Output Estimator Annotation

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    @inlinable public func update InputSequence inout
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator  Transformer           InputSequence
Event Handler  nil  async throws where
InputSequence Sequence InputSequence Element
AnnotatedFeature Preprocessor Input Estimator Annotation

    /// Encodes a fitted transformer.
    public func encode inout any
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator  Transformer           inout any
EstimatorEncoder throws

```

```

    /// Decodes a previously fitted transformer.
    public func decode inout any
EstimatorDecoder throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
    public func encodeWithOptimizer _
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer inout any
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    public func decodeWithOptimizer inout any
EstimatorDecoder throws
PreprocessingUpdatableSupervisedEstimator Preprocessor
Estimator Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension PreprocessingUpdatableSupervisedEstimator Sendable
where Preprocessor Sendable Estimator Sendable

    /// An updatable supervised estimator that composes a preprocessing transformer
    /// and an updatable supervised estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator UpdatableSupervisedTabularEstimator where
Preprocessor TabularTransformer Estimator
UpdatableSupervisedTabularEstimator

    /// The transformer type created by this estimator.
    public typealias Transformer
ComposedTabularTransformer Preprocessor
Estimator Transformer

    /// The input type.
    public typealias Input Preprocessor Input

```

```
    ///> The intermediate type.
    public typealias Intermediate Preprocessor Output

    ///> The output type.
    public typealias Output Estimator Transformer Output

    ///> The annotation type.
    public typealias Annotation Estimator Annotation

    ///> The preprocessing transformer.
    public var preprocessor Preprocessor

    ///> The estimator.
    public var estimator Estimator

    ///> The annotation column identifier.
    public var annotationColumnID ColumnID Estimator Annotation

    ///> Creates a composed supervised estimator from a preprocessing
    transformer and a supervised estimator.
    public init _ Preprocessor _ Estimator

    ///> Creates a default-initialized transformer suitable for incremental fitting.
    @inlinable public func makeTransformer
    PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
    Estimator Transformer

    ///> Prepares a data frame.
    ///
    ///> - Parameters:
    ///>   - input: A data frame of examples.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The preprocessed examples.
    @inlinable public func preprocessed DataFrame
        EventHandler nil async throws DataFrame

    ///> Fits a composed transformer to a data frame of examples.
    ///
    ///> - Parameters:
    ///>   - preprocessedInput: A data frame of preprocessed features
    used for fitting the transformer.
    ///>   - preprocessedValidation: A data frame of preprocessed
    features used for validating the fitted transformer.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
    @inlinable public func fitted DataFrame
        DataFrame
```

```

EventHandler    nil  async throws
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator   Transformer

    /// Fits a composed transformer to a data frame of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples used for fitting the
    transformer.
    ///   - validation: A data frame containing examples used for
    validating the fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted           DataFrame
                                         DataFrame          EventHandler
    nil  async throws
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator   Transformer

    /// Updates a transformer with a new data frame of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A data frame of preprocessed features.
    ///   - eventHandler: An event handler.
    @inlinable public func update _           inout
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator   Transformer
DataFrame          EventHandler      nil  async throws

    /// Updates a transformer with a new data frame of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A data frame of examples.
    ///   - eventHandler: An event handler.
    @inlinable public func update _           inout
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator   Transformer          DataFrame
EventHandler      nil  async throws

    /// Encodes a fitted transformer.
    public func encode _
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator   Transformer          inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws

```

```
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator Transformer
```

```
    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
    public func encodeWithOptimizer _  
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator Transformer           inout any  
EstimatorEncoder throws  
  
    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    public func decodeWithOptimizer           inout any  
EstimatorDecoder throws  
PreprocessingUpdatableSupervisedTabularEstimator Preprocessor
Estimator Transformer
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension PreprocessingUpdatableSupervisedTabularEstimator
Sendable where Preprocessor Sendable Estimator Sendable
```

```
/// An updatable supervised temporal estimator that composes a preprocessing
transformer and an updatable supervised temporal estimator.
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
public struct
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator UpdatableSupervisedTemporalEstimator where
Preprocessor TemporalTransformer Estimator
UpdatableSupervisedTemporalEstimator Preprocessor Output
Estimator Transformer Input
```

```
    /// The transformer type created by this estimator.
    public typealias Transformer
ComposedTemporalTransformer Preprocessor
Estimator Transformer
```

```
    /// The input type.
    public typealias Input   Preprocessor Input
```

```

    /// The intermediate type.
public typealias Intermediate Preprocessor Output

    /// The output type.
public typealias Output Estimator Transformer Output

    /// The annotation type.
public typealias Annotation Estimator Annotation

    /// The preprocessing transformer.
public var preprocessor Preprocessor

    /// The estimator.
public var estimator Estimator

    /// Creates a composed supervised temporal estimator from a preprocessing
transformer and a supervised temporal estimator.
public init _ Preprocessor _ Estimator

    /// Prepares a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The preprocessed examples.
    @inlinable public func preprocessed InputSequence
FeatureSequence InputSequence
EventHandler nil async throws
AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out
put
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Annotation where InputSequence Sequence
FeatureSequence TemporalSequence Preprocessor Input
FeatureSequence Feature InputSequence Element
AnnotatedFeature FeatureSequence Estimator Annotation

    /// Fits a transformer to a sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted
AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out
put
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Annotation EventHandler nil
async throws

```

```
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor  
Estimator Transformer
```

```
    /// Fits a transformer to a sequence of preprocessed features while  
    validating.  
    ///  
    /// - Parameters:  
    /// - preprocessedInput: A sequence of preprocessed features.  
    /// - preprocessedValidation: A sequence of preprocessed  
    features used for validating the fitted transformer.  
    /// - eventHandler: An event handler.  
    /// - Returns: The fitted transformer.  
    @inlinable public func fitted  
  
    AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out  
put  
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor  
Estimator Annotation  
    AnnotatedFeature PreprocessedFeatureSequence Preprocessor Out  
put  
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor  
Estimator Annotation EventHandler nil  
async throws  
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor  
Estimator Transformer  
  
    /// Fits a transformer to a sequence of examples.  
    ///  
    /// - Parameters:  
    /// - input: A sequence of examples used for fitting the transformer.  
    /// - eventHandler: An event handler.  
    /// - Returns: The fitted transformer.  
    @inlinable public func fitted InputSequence  
FeatureSequence InputSequence  
EventHandler nil async throws  
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor  
Estimator Transformer where InputSequence Sequence  
FeatureSequence TemporalSequence Preprocessor Input  
FeatureSequence Feature InputSequence Element  
AnnotatedFeature FeatureSequence Estimator Annotation  
  
    /// Fits a transformer to a sequence of examples while validating with a  
validation sequence.  
    ///  
    /// - Parameters:  
    /// - input: A sequence of examples used for fitting the transformer.  
    /// - validation: A sequence of examples used for validating the  
fitted transformer.  
    /// - eventHandler: An event handler.  
    /// - Returns: The fitted transformer.
```

```

@inlinable public func fitted InputSequence Validation
FeatureSequence InputSequence
Validation EventHandler nil
async throws
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Transformer where InputSequence Sequence
Validation Sequence FeatureSequence TemporalSequence
Preprocessor Input FeatureSequence Feature
InputSequence Element AnnotatedFeature FeatureSequence
Estimator Annotation Validation Element
AnnotatedFeature FeatureSequence Estimator Annotation

    /// Creates a default-initialized transformer suitable for incremental fitting.
@inlinable public func makeTransformer
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Transformer

    /// Updates a transformer with a new sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
@inlinable public func update InputSequence
FeatureSequence _ inout
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Transformer
InputSequence EventHandler nil async throws
where InputSequence Sequence FeatureSequence
TemporalSequence Preprocessor Output
FeatureSequence Feature InputSequence Element
AnnotatedFeature FeatureSequence Estimator Annotation
FeatureSequence Feature Estimator Transformer Input

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
@inlinable public func update InputSequence
FeatureSequence _ inout
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
Estimator Transformer InputSequence
EventHandler nil async throws where
InputSequence Sequence FeatureSequence TemporalSequence
Preprocessor Input FeatureSequence Feature
InputSequence Element AnnotatedFeature FeatureSequence
Estimator Annotation

```

```

    /// Encodes a fitted transformer.
    public func encode_
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
    Estimator Transformer           inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode             inout any
EstimatorDecoder throws
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
    Estimator Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
    public func encodeWithOptimizer_
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
    Estimator Transformer           inout any
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    public func decodeWithOptimizer           inout any
EstimatorDecoder throws
PreprocessingUpdatableSupervisedTemporalEstimator Preprocessor
    Estimator Transformer

```

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		
extension PreprocessingUpdatableSupervisedTemporalEstimator		
Sendable where Preprocessor Sendable Estimator Sendable		

```

    /// An updatable estimator that composes a preprocessing transformer and an
    updatable estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct
PreprocessingUpdatableTabularEstimator Preprocessor
    Estimator UpdatableTabularEstimator where Preprocessor
    TabularTransformer Estimator UpdatableTabularEstimator

```

```
    ///> /// The transformer type created by this estimator.
    public typealias Transformer
ComposedTabularTransformer Preprocessor
Estimator Transformer

    ///> /// The input type.
    public typealias Input    Preprocessor Input

    ///> /// The intermediate type.
    public typealias Intermediate    Preprocessor Output

    ///> /// The output type.
    public typealias Output    Estimator Transformer Output

    ///> /// The preprocessing transformer.
    public var preprocessor    Preprocessor

    ///> /// The estimator.
    public var estimator    Estimator

    ///> /// Creates a composed updatable estimator from a preprocessing
transformer and an estimator.
    public init _           Preprocessor _           Estimator

    ///> /// Creates a default-initialized transformer suitable for incremental fitting.
    @inlinable public func makeTransformer
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer

    ///> /// Preprocesses a data frame of examples.
    ///
    ///> /// - Parameters:
    ///>   - input: A data frame of examples.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The preprocessed examples.
    @inlinable public func preprocessed      DataFrame
        EventHandler    nil  async throws      DataFrame

    ///> /// Fits a transformer to a data frame of preprocessed features.
    ///
    ///> /// - Parameters:
    ///>   - preprocessed: A data frame of preprocessed features.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
    @inlinable public func fitted
DataFrame          EventHandler    nil  async throws
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer
```

```

    /// Fits a composed transformer to a data frame of examples.
    ///
    /// - Parameters:
    ///   - input: A data frame of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted           DataFrame
                                EventHandler nil async throws
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer

    /// Updates a transformer with a new data frame of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A data frame of preprocessed features.
    ///   - eventHandler: An event handler.
    @inlinable public func update _          inout
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer
DataFrame           EventHandler nil async throws

    /// Updates a transformer with a new data frame of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A data frame of examples.
    ///   - eventHandler: An event handler.
    @inlinable public func update _          inout
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer           DataFrame
EventHandler nil async throws

    /// Encodes a fitted transformer.
    public func encode _
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer           inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws
PreprocessingUpdatableTabularEstimator Preprocessor
Estimator Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:

```

```

    /// - transformer: A transformer this estimator creates.
    /// - encoder: An encoder.
public func encodeWithOptimizer _  

PreprocessingUpdatableTabularEstimator Preprocessor  

Estimator Transformer inout any  

EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer.
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
public func decodeWithOptimizer inout any  

EstimatorDecoder throws  

PreprocessingUpdatableTabularEstimator Preprocessor  

Estimator Transformer

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  

extension PreprocessingUpdatableTabularEstimator Sendable  

where Preprocessor Sendable Estimator Sendable

/// An updatable temporal estimator that composes a preprocessing transformer
and an updatable temporal estimator.
available 13.0 15.0  

available 16.0 18.0  

available 16.0 18.0  

available 1.0 2.0  

available  

public struct  

PreprocessingUpdatableTemporalEstimator Preprocessor  

Estimator UpdatableTemporalEstimator where Preprocessor  

TemporalTransformer Estimator UpdatableTemporalEstimator  

Preprocessor Output Estimator Transformer Input

    /// The transformer type created by this estimator.
public typealias Transformer  

ComposedTemporalTransformer Preprocessor  

Estimator Transformer

    /// The input type.
public typealias Input Preprocessor Input

    /// The intermediate type.
public typealias Intermediate Preprocessor Output

    /// The output type.
public typealias Output Estimator Transformer Output

```

```
    ///> The preprocessing transformer.  
    public var preprocessor Preprocessor  
  
    ///> The estimator.  
    public var estimator Estimator  
  
    ///> Creates a composed temporal estimator from a preprocessing  
    transformer and a temporal estimator.  
    public init _ Preprocessor _  
Estimator  
  
    ///> Preprocesses a sequence of examples.  
    ///>  
    ///> - Parameters:  
    ///>   - input: A sequence of examples.  
    ///>   - eventHandler: An event handler.  
    ///> - Returns: The preprocessed examples.  
    @inlinable public func preprocessed InputSequence  
        InputSequence EventHandler nil async  
throws PreprocessedFeatureSequence Preprocessor Output  
where InputSequence Sequence Preprocessor Input  
InputSequence Element Feature InputSequence Element  
TemporalSequence  
  
    ///> Fits a transformer to a sequence of preprocessed features.  
    ///>  
    ///> - Parameters:  
    ///>   - preprocessed: A sequence of preprocessed features.  
    ///>   - eventHandler: An event handler.  
    ///> - Returns: The fitted transformer.  
    @inlinable public func fitted  
        PreprocessedFeatureSequence Preprocessor Output  
            EventHandler nil async throws  
PreprocessingUpdatableTemporalEstimator Preprocessor  
Estimator Transformer  
  
    ///> Fits a transformer to a sequence of examples.  
    ///>  
    ///> - Parameters:  
    ///>   - input: A sequence of examples.  
    ///>   - eventHandler: An event handler.  
    ///> - Returns: The fitted transformer.  
    @inlinable public func fitted InputSequence  
        InputSequence EventHandler nil async throws  
PreprocessingUpdatableTemporalEstimator Preprocessor  
Estimator Transformer where InputSequence Sequence  
Preprocessor Input InputSequence Element Feature  
InputSequence Element TemporalSequence
```

```
    /// Creates a default-initialized transformer suitable for incremental fitting.
    @inlinable public func makeTransformer
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer

    /// Updates a transformer with a new sequence of preprocessed features.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - preprocessed: A sequence of preprocessed features.
    ///   - eventHandler: An event handler.
    @inlinable public func update InputSequence _ inout
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer
InputSequence           EventHandler nil async throws
where InputSequence Sequence InputSequence Element
TemporalSequence Estimator Transformer Input
InputSequence Element Feature

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    @inlinable public func update InputSequence _ inout
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer           InputSequence
                                EventHandler nil async throws where
InputSequence Sequence Preprocessor Input
InputSequence Element Feature InputSequence Element
TemporalSequence

    /// Encodes a fitted transformer.
    public func encode _ inout any
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer           inout any
EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    public func decode           inout any
EstimatorDecoder throws
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
```

```

    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
public func encodeWithOptimizer _
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer inout any
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
public func decodeWithOptimizer inout any
EstimatorDecoder throws
PreprocessingUpdatableTemporalEstimator Preprocessor
Estimator Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension PreprocessingUpdatableTemporalEstimator Sendable
where Preprocessor Sendable Estimator Sendable

    /// Crops an image at a random location.
available macOS 14.0 iOS 17.0 tvOS 17.0
public struct RandomImageCropper RandomTransformer Sendable

    /// Creates an augmentation that crops an input image at a random location
    to the specified target size.
    ///
    /// - Parameter targetSize: The target size of the cropping rectangle.
    Must be positive.
public init CGSize

    /// Creates an augmentation that crops an input image at a random location
    to the specified target width and
    /// height.
    ///
    /// - Parameters:
    ///   - targetWidth: The target width of the cropping rectangle. Must be
    positive.
    ///   - targetHeight: The target height of the cropping rectangle. Must
    be positive.
public init Double Double

```

```
    /// Creates an augmentation that crops an input image at a random location
    /// with a scale that indicates the lower
    /// and upper bounds to randomly scale the height and width of the image.
    /// The range must be between 0 and 1.
    ///
    /// - Parameters:
    ///   - scale: A range of scales.
    ///   - aspectRatio: A size that specifies the ratio of width to height to
    ///     use for the cropping rectangle.
    public init      ClosedRange Double
Double      nil

    /// Randomly crops an image at a random location of a given size.
    /// - Parameters:
    ///   - image: The input image.
    ///   - generator: A random number generator.
    ///   - eventHandler: An event handler.
    /// - Returns: The cropped image.
    public func applied      CIImage      inout
some RandomNumberGenerator      EventHandler      nil
async throws      CIImage

    /// The input type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Input      CIImage

    /// The output type.
@available iOS 17.0 tvOS 17.0 macOS 14.0
public typealias Output      CIImage

    /// A transformer that adds random noise to an image.
@available macOS 14.0 iOS 17.0 tvOS 17.0
public struct RandomImageNoiseGenerator      Transformer
Sendable

    /// The intensity of the random noise to add to the image.
public var intensity  Double

    /// Creates transformer that generates random noise to apply to an image.
    ///
    /// - Parameter intensity: The intensity of the random noise to add to
    /// the image.
    public init      Double

    /// Adds random noise to the input image.
    ///
    /// - Parameters:
    ///   - image: An image.
    ///   - eventHandler: An event handler.
```

```

    /// - Returns: The image with added random noise.
    public func applied inout some CIImage
EventHandler nil CIImage

    /// The input type.
    @available iOS 17.0 tvOS 17.0 macOS 14.0
    public typealias Input CIImage

    /// The output type.
    @available iOS 17.0 tvOS 17.0 macOS 14.0
    public typealias Output CIImage

    /// A transformer that takes an input and a random number generator and
    produces a randomized output.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
    public protocol RandomTransformer

        /// The input type.
        associatedtype Input

        /// The output type.
        associatedtype Output

        /// Performs the random transformation on a single input.
        ///
        /// - Parameters:
        ///   - input: The random transformer input.
        ///   - generator: A random number generator.
        ///   - eventHandler: An event handler.
        /// - Returns: An output produced by applying the random transformer to
        the input.
        func applied Self Input inout some EventHandler async
RandomNumberGenerator throws Self Output

    /// A transformer that predicts a float value.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public protocol Regressor Transformer where Self Output FloatingPoint

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    extension Regressor

        /// Performs a prediction from a single input.
        ///
        /// - Parameter input: The input feature.
        /// - Returns: A regression.

```

```

public func prediction           Self Input  async
throws      Self Output

    /// Performs a prediction from a sequence of inputs.
    ///
    /// - Parameter input: The input features.
    /// - Returns: An array of regressions.
public func prediction S           S async throws
Self Output  where S Sequence  Self Input  S Element

/// A transformer that reshapes a shaped array.
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public struct Reshaper Scalar  Transformer Equatable
 Codable where Scalar  MLShapedArrayScalar  Scalar
 Decodable  Scalar  Encodable

    /// The target shape.
public var shape  Int

    /// Creates a reshape transformer.
    ///
    /// - Parameters:
    ///   - shape: The target shape.
public init          Int

    /// Reshapes the input.
    ///
    /// - Parameters:
    ///   - input: A shaped array.
    ///   - eventHandler: An event handler.
    /// - Returns: A shaped array with the target shape.
public func applied          MLShapedArray Scalar
                    EventHandler  nil throws
MLShapedArray Scalar

    /// Reshapes a sequence of inputs.
    ///
    /// - Parameters:
    ///   - input: A sequence of input shaped arrays.
    ///   - eventHandler: An event handler.
    /// - Returns: An array of shaped arrays.
public func applied S -      S
EventHandler  nil throws  MLShapedArray Scalar  where S
Sequence  S Element  MLShapedArray Scalar

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,

```

```

    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func Reshape Scalar
Reshape Scalar Bool

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Input MLShapedArray Scalar

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

public typealias Output MLShapedArray Scalar

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Reshape CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`
    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///

```

```

    /**
     struct Point: CustomDebugStringConvertible {
     let x: Int, y: Int
     /**
      var debugDescription: String {
      return "(\(x), \(y))"
     }
     */
     let p = Point(x: 21, y: 30)
     let s = String(reflecting: p)
     print(s)
     // Prints "(21, 30)"
     /**
     The conversion of `p` to a string in the assignment to `s` uses the
     `Point` type's `debugDescription` property.
  public var debugDescription String get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Reshaper Sendable where Scalar Sendable

```

```

    /**
     An estimator that scales the input using statistics that are robust to outliers.
  @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
  public struct RobustScaler Element Estimator where
  Element BinaryFloatingPoint Element Decodable Element
  Encodable

    /**
     The quantile range used to compute the scale.
  public var quantileRange ClosedRange Element

    /**
     Creates a robust scaler.
    /**
     - Parameter quantileRange: This scaler removes the median and
     scales the data according to the quantile range.
  public init ClosedRange Element
  0.25 0.75

    /**
     Fits a robust scaler to a sequence of elements.
    /**
    /**
     - Parameters:
     - input: A sequence of elements.
     - eventHandler: An event handler.
    /**
     - Returns: The fitted transformer.
  public func fitted S S
  EventHandler nil throws
  RobustScaler Element Transformer where Element S Element
  S Sequence

```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension RobustScaler Sendable where Element Sendable

extension RobustScaler

    /// A transformer that scales the input using statistics that are robust to outliers.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public struct Transformer : Transformer, Hashable

        /// The median used for offsetting.
        public var median : Element

        /// The interquartile range used for scaling.
        public var interQuartileRange : Element

        /// Creates a robust scaling transformer.
        ///
        /// - Parameters:
        ///   - median: The median used for offsetting.
        ///   - interQuartileRange: The inter quartile range used for scaling.
        public init (Element, Element)

        /// Scales the input values using the calculation `(input - median) / interQuartileRange`.
        ///
        /// - Parameters:
        ///   - input: A floating-point value.
        ///   - eventHandler: An event handler.
        /// - Returns: A scaled value.
        @inlinable public func applied (EventHandler, nil, Element)

        /// Hashes the essential components of this value by feeding them into the given hasher.
        ///
        /// Implement this method to conform to the `Hashable` protocol. The components used for hashing must be the same as the components compared
        /// in your type's `==` operator implementation. Call `hasher.combine(_:)` with each of these components.
        ///
        /// - Important: In your implementation of `hash(into:)`, don't call `finalize()` on the `hasher` instance provided,
```

```

    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
RobustScaler Element Transformer
RobustScaler Element Transformer Bool

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
public typealias Input Element

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
public typealias Output Element

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension RobustScaler Transformer
CustomDebugStringConvertible

```

```

/// A textual representation of this instance, suitable for debugging.
///
/// Calling this property directly is discouraged. Instead, convert an
/// instance of any type to a string by using the `String(reflecting:)` initializer.
/// This initializer works with any type, and uses the custom
/// `debugDescription` property for types that conform to
/// `CustomDebugStringConvertible`:
///
///     struct Point: CustomDebugStringConvertible {
///         let x: Int, y: Int
///
///         var debugDescription: String {
///             return "(\(x), \(y))"
///         }
///     }
///
///     let p = Point(x: 21, y: 30)
///     let s = String(reflecting: p)
///     print(s)
///     // Prints "(21, 30)"
///
/// The conversion of `p` to a string in the assignment to `s` uses the
/// `Point` type's `debugDescription` property.
public var debugDescription String get

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension RobustScaler Transformer Encodable

```

/// Encodes this value into the given encoder.
///
/// If the value fails to encode anything, `encoder` will encode an empty
/// keyed container in its place.
///
/// This function throws an error if any values are invalid for the given
/// encoder's format.
///
/// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension RobustScaler Transformer Decodable

```

/// Creates a new instance by decoding from the given decoder.
///
/// This initializer throws an error if reading from the decoder fails, or
/// if the data read is corrupted or otherwise invalid.

```

```
///  
/// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension RobustScaler Transformer Sendable where Element  
Sendable  
  
/// A serialization error.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public enum SerializationError : LocalizedError, Equatable  
Sendable  
  
/// An error that indicates that the package already exists at the URL.  
case packageAlreadyExists URL  
  
/// An error that indicates that the package at specified URL was not found.  
case packageNotFound URL  
  
/// An error that indicates that the transformer cannot be represented as a  
CoreML model.  
case notRepresentableAsCoreML String  
  
/// A localized message describing what error occurred.  
public var errorMessage String get  
  
/// Returns a Boolean value indicating whether two values are equal.  
///  
/// Equality is the inverse of inequality. For any values `a` and `b`,  
/// `a == b` implies that `a != b` is `false`.  
///  
/// - Parameters:  
///   - lhs: A value to compare.  
///   - rhs: Another value to compare.  
public static func SerializationError  
SerializationError Bool  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension SerializationError : CustomDebugStringConvertible  
  
/// A text representation of the error.  
public var debugDescription String get  
  
/// Apply transformations in a random order.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
```

```
public struct ShuffleRandomly Element RandomTransformer

    /// Creates a random shuffle augmentation.

    /// - Parameters:
    ///   - augmentation: An augmentation builder.

    public init RandomTransformer AugmentationBuilder Element _ RandomTransformer where Element == RandomTransformer Input RandomTransformer RandomTransformer RandomTransformer Input RandomTransformer Output

    /// Apply transformations in a random order.

    /// - Parameters:
    ///   - input: The input to the transformer.
    ///   - generator: A random number generator.
    ///   - eventHandler: An event handler.

    /// - Returns: The augmented input.

    public func applied Element inout some RandomNumberGenerator EventHandler nil async throws Element

    /// The input type.

    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Input Element

    /// The output type.

    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Output Element


    /// A temporal transformer that groups input elements.

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public struct SlidingWindowTransformer Input TemporalTransformer Codable Sendable where Input == Sendable

    /// The input type.

    public typealias Input Input

    /// The output type.

    public typealias Output Input

    /// The number of elements between the start of two consecutive windows.

    public let stride Int
```

```
    /// The length of a window.
public let length Int

    /// Creates a window temporal transformer.
    /// - Parameters:
    ///   - stride: The number of frames between the start of two consecutive windows. Must be greater than 0.
    ///   - length: The length of a window in number of frames. Must be greater than 0.
public init Int Int

    /// Extracts a window sequence from the input sequence
    ///
    /// - Parameters:
    ///   - input: An async sequence of inputs.
    ///   - eventHandler: An event handler.
    /// - Returns: An async sequence of windowed outputs.
public func applied S S
EventHandler throws
SlidingWindowTransformer Input WindowSequence where Input S Feature S TemporalSequence

    /// The output async sequence type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias OutputSequence
SlidingWindowTransformer Input WindowSequence

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SlidingWindowTransformer
```

```

    /// An async sequence of windows.
public struct WindowSequence : TemporalSequence

    /// The feature type.
public typealias Feature = Input

    /// The type of asynchronous iterator that produces elements of this
    /// asynchronous sequence.
public typealias AsyncIterator = SlidingWindowTransformer<Input, WindowSequence> Iterator

    /// The number of elements in the sequence.
public var count: Int { get }

    /// Creates the asynchronous iterator that produces elements of this
    /// asynchronous sequence.
    ///
    /// - Returns: An instance of the `AsyncIterator` type used to
produce
    /// elements of the asynchronous sequence.
public func makeAsyncIterator() SlidingWindowTransformer<Input, WindowSequence> Iterator

    /// The type of element produced by this asynchronous sequence.
@available(iOS 16.0, tvOS 16.0, watchOS 11.0, macOS 13.0)
public typealias Element = TemporalFeature<SlidingWindowTransformer<Input, WindowSequence>, Feature>

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
extension SlidingWindowTransformer<WindowSequence>

    /// An async iterator of window sequence.
public struct Iterator : AsyncIteratorProtocol

        /// Asynchronously advances to the next element and returns it, or
ends the
        /// sequence if there is no next element.
        ///
        /// - Returns: The next element, if it exists, or `nil` to signal the
end of
        /// the sequence.
public mutating func next() async throws TemporalFeature<SlidingWindowTransformer<Input, Output>>

@available(iOS 16.0, tvOS 16.0, watchOS 11.0, macOS

```

13.0

```
public typealias Element
TemporalFeature SlidingWindowTransformer Input Output
```

```
/// A sequence of windows on a time series shaped array.
///
/// The shape of each window in the sequence is `[length, featureSize]`.
/// The sequence will return as many windows as
/// fit in the input. For example, an input shaped array of shape `[8, 1]` using
/// `stride` of 1 and `length` of 4 will
/// produce 5 examples:
///
/// ````
/// [[1], [2], [3], [4]]
/// [[2], [3], [4], [5]]
/// [[3], [4], [5], [6]]
/// [[4], [5], [6], [7]]
/// [[5], [6], [7], [8]]
/// ````
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct SlidingWindows : RandomAccessCollection
@unchecked Sendable where Scalar : MLShapedArrayScalar

/// The input shaped array.
public let input : MLShapedArray

/// The number samples in each window.
public let length : Int

/// The number of samples between windows.
public let stride : Int

/// The position of the first window.
public var startIndex : Int get

/// The collection's "past the end" position—that is, the position one greater
/// than the last valid subscript
/// argument.
public var endIndex : Int get

/// Creates a sliding windows sequence.
///
/// - Parameters:
///   - input: A shaped array having two dimensions.
///   - length: The number of samples in each window. Must be
///     positive.
///   - stride: The number of samples between windows. Must be
```

```

positive. Defaults to 1.
public init          MLShapedArray Scalar           Int
    Int  1 throws

    /// Returns the position immediately before the given index.
    ///
    /// - Parameter i: A valid index of the collection. `i` must be greater
than `startIndex`.
    /// - Returns: The index value immediately before `i`.
public func index      Int      Int

    /// Returns the position immediately after the given index.
    ///
    /// - Parameter i: A valid index of the collection. `i` must be less than
`endIndex`.
    /// - Returns: The index value immediately after `i`.
public func index      Int      Int

    /// Returns an index that is the specified distance from the given index.
    ///
    /// The value passed as `distance` must not offset `i` beyond the
bounds of the collection.
    ///
    /// - Parameters:
    ///     - i: A valid index of the collection.
    ///     - distance: The distance to offset `i`.
    /// - Returns: An index offset by `distance` from the index `i`.
public func index _     Int      Int      Int

    /// Accesses the window at the specified position.
    ///
    /// - Parameter position: The position of the window to access.
`position` must be a valid index of the
    /// collection that is not equal to the `endIndex` property.
public subscript        Int      MLShapedArray Scalar
get

    /// Accesses a contiguous range of windows.
    ///
    /// - Parameter bounds: A range of valid indices in the classification
distribution.
public subscript        Range Int
Slice SlidingWindows Scalar   get

    /// A type representing the sequence's elements.
@available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS
2.0 macOS 15.0
public typealias Element MLShapedArray Scalar

    /// A type that represents a position in the collection.

```

```

    /**
     * Valid indices consist of the position of every element and a
     * "past the end" position that's not valid for use as a subscript
     * argument.
     */
    @available(iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0)
    public typealias Index = Int

    /**
     * A type that represents the indices that are valid for subscripting the
     * collection, in ascending order.
     */
    @available(iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0)
    public typealias Indices = Range<Int>

    /**
     * A type that provides the collection's iteration interface and
     * encapsulates its iteration state.
     */
    /**
     * By default, a collection conforms to the `Sequence` protocol by
     * supplying `IndexingIterator` as its associated `Iterator`
     * type.
     */
    @available(iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0)
    public typealias Iterator = IndexingIterator<SlidingWindows<Scalar>>

    /**
     * A collection representing a contiguous subrange of this collection's
     * elements. The subsequence shares indices with the original collection.
     */
    /**
     * The default subsequence type for collections that don't define their own
     * is `Slice`.
     */
    @available(iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0)
    public typealias SubSequence = Slice<SlidingWindows<Scalar>>

    /**
     * An estimator that standardizes the input by removing the mean and scaling to
     * unit variance.
     */
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    public struct StandardScaler<Element: BinaryFloatingPoint, Element: Decodable, Element: Encodable> {
        /**
         * Creates a standard scaling estimator.
         */
        public init()
        /**
         * Fits a transformer to a particular input sequence by computing the mean
         * and standard deviation.
         */
        public func fitted<S>(EventHandler<nil> throws

```

```
StandardScaler Element Transformer where Element
S Element S Sequence
```

```
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
extension StandardScaler UpdatableEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func makeTransformer
    StandardScaler Element Transformer

        /// Updates a transformer with a new sequence of examples.
        ///
        /// - Note: You can't add new categories on subsequent updates. All
        categories should be present in the initial
        /// update.
        ///
        /// - Parameters:
        ///     - transformer: A transformer to update.
        ///     - input: A sequence of examples.
        ///     - eventHandler: An event handler.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func update _ inout
    StandardScaler Element Transformer some
    Sequence Element EventHandler nil

        /// Encodes the transformer to an encoder.
        ///
        /// - Parameters:
        ///     - transformer: A transformer this estimator creates.
        ///     - encoder: An encoder.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func encodeWithOptimizer _ inout any
    StandardScaler Element Transformer EstimatorEncoder throws

        /// Reads the encoded transformer with a decoder.
        ///
        /// - Parameter decoder: A decoder.
        /// - Returns: The decoded transformer.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func decodeWithOptimizer inout any
    EstimatorDecoder throws
    StandardScaler Element Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension StandardScaler Sendable where Element Sendable

extension StandardScaler

    /// A transformer that standardizes the input by removing the mean and
    scaling to unit variance.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public struct Transformer : Transformer, Hashable

        /// The mean used for offsetting.
        public var mean : Element

        /// The standard deviation used for scaling.
        public var standardDeviation : Element

        /// Creates a standard scaling transformer.
        ///
        /// - Parameters:
        ///   - mean: The mean used for offsetting.
        ///   - standardDeviation: The standard deviation used for
scaling.
        public init Element Element

            /// Scales the input values using the calculation `(input - mean) /
standardDeviation`.
            ///
            /// - Parameters:
            ///   - input: A floating-point value.
            ///   - eventHandler: An event handler.
            /// - Returns: A scaled value.
@inlinable public func applied Element
EventHandler nil Element

            /// Hashes the essential components of this value by feeding them into
the
            /// given hasher.
            ///
            /// Implement this method to conform to the `Hashable` protocol. The
            /// components used for hashing must be the same as the components
compared
            /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
            /// with each of these components.
            ///
            /// - Important: In your implementation of `hash(into:)`,
```

```

    /// don't call `finalize()` on the `hasher` instance provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
StandardScaler Element Transformer
StandardScaler Element Transformer Bool

    /// The input type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
public typealias Input Element

    /// The output type.
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS
13.0
public typealias Output Element

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for
you.
public var hashValue Int get


```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension StandardScaler Transformer

```

CustomDebugStringConvertible

```
/// A textual representation of this instance, suitable for debugging.  
///  
/// Calling this property directly is discouraged. Instead, convert an  
/// instance of any type to a string by using the `String(describing:)`  
/// initializer. This initializer works with any type, and uses the custom  
/// `debugDescription` property for types that conform to  
/// `CustomDebugStringConvertible`:  
///  
///     struct Point: CustomDebugStringConvertible {  
///         let x: Int, y: Int  
///  
///         var debugDescription: String {  
///             return "(\(x), \(y))"  
///         }  
///     }  
///  
///     let p = Point(x: 21, y: 30)  
///     let s = String(describing: p)  
///     print(s)  
///     // Prints "(21, 30)"  
///  
/// The conversion of `p` to a string in the assignment to `s` uses the  
/// `Point` type's `debugDescription` property.  
public var debugDescription String get
```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension StandardScaler Transformer Encodable

```
/// Encodes this value into the given encoder.  
///  
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.  
///  
/// This function throws an error if any values are invalid for the given  
/// encoder's format.  
///  
/// - Parameter encoder: The encoder to write data to.  
public func encode any Encoder throws
```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension StandardScaler Transformer Decodable

```
/// Creates a new instance by decoding from the given decoder.  
///  
/// This initializer throws an error if reading from the decoder fails, or
```

```

    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension StandardScaler Transformer Sendable where
Element Sendable

    /// An estimator that creates a transformer by fitting to a data set.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol SupervisedEstimator

    /// The transformer type created by this estimator.
associatedtype Transformer Transformer

    /// The annotation type.
associatedtype Annotation Equatable

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
func fitted Input Input
EventHandler async throws Self Transformer where Input
Sequence Input Element
AnnotatedFeature Self Transformer Input Self Annotation

    /// Fits a transformer to a sequence of examples while validating with a
    validation sequence.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
    fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
func fitted Input Validation Input
Validation EventHandler async
throws Self Transformer where Input Sequence
Validation Sequence Input Element
AnnotatedFeature Self Transformer Input Self Annotation
Validation Element AnnotatedFeature Self Transformer Input
Self Annotation

    /// Encodes a fitted transformer.

```

```

func encode _ Self Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
func decode inout any EstimatorDecoder
throws Self Transformer

extension SupervisedEstimator

    /// Composes this supervised estimator with a temporal transformer.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
@preconcurrency public func appending Other _
Other some
SupervisedTemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer Other
Self Annotation where Other TemporalTransformer
Self Annotation Sendable Other Input
Self Transformer Output

    /// Composes this supervised estimator with a temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
@preconcurrency public func appending Other _
Other some
SupervisedTemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer Other Transformer
Self Annotation where Other TemporalEstimator
Self Annotation Sendable Self Transformer Output
Other Transformer Input

    /// Composes this supervised estimator with a supervised temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other some
SupervisedTemporalEstimator ComposedTemporalTransformer Transf

```

```
ormerToTemporalAdaptor Self Transformer Other Transformer
Self Annotation where Other SupervisedTemporalEstimator
Self Annotation Other Annotation Self Transformer Output
Other Transformer Input
```

```
extension SupervisedEstimator where Self Annotation Sendable
```

```
/// Exposes this supervised estimator as a temporal supervised estimator.
```

```
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
```

```
    @inlinable public func adaptedAsTemporal
```

```
SupervisedEstimatorToTemporalAdaptor Self
```

```
extension SupervisedEstimator
```

```
/// Composes this supervised estimator with a transformer.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
    public func appending Other _ Other some
```

```
SupervisedEstimator ComposedTransformer Self Transformer
Other Self Annotation where Other Transformer
Other Input Self Transformer Output
```

```
/// Composes this supervised estimator with an estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
    public func appending Other _ Other some
```

```
SupervisedEstimator ComposedTransformer Self Transformer
Other Transformer Self Annotation where Other Estimator
Self Transformer Output Other Transformer Input
```

```
/// Composes this supervised estimator with another supervised estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
    public func appending Other _ Other some
```

```
SupervisedEstimator ComposedTransformer Self Transformer
Other Transformer Self Annotation where Other
SupervisedEstimator Self Annotation Other Annotation
Self Transformer Output Other Transformer Input
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SupervisedEstimator where Self Transformer
Encodable
```

```
    /// Encodes a fitted encodable transformer.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - encoder: An estimator encoder.
public func encode _ Self Transformer
inout any EstimatorEncoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SupervisedEstimator where Self Transformer
Decodable
```

```
    /// Decodes a previously fitted decodable transformer.
    ///
    /// - Parameter decoder: An estimator decoder.
    /// - Returns: The decoded transformer.
public func decode inout any
EstimatorDecoder throws Self Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SupervisedEstimator
```

```
    /// Writes the encoded transformer to a file.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - url: A file URL.
    ///   - overwrite: A Boolean value indicating whether to overwrite
existing files.
public func write _ Self Transformer
URL Bool true throws

    /// Reads the encoded transformer from a file.
    ///
    /// - Parameter url: A file URL.
    /// - Returns: The decoded transformer.
public func read URL throws Self Transformer
```

```
extension SupervisedEstimator
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func fitted Input Input
async throws Self Transformer where Input Sequence
Input Element AnnotatedFeature Self Transformer Input
Self Annotation

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func fitted Input Validation
Input Validation async throws
Self Transformer where Input Sequence Validation
Sequence Input Element
AnnotatedFeature Self Transformer Input Self Annotation
Validation Element AnnotatedFeature Self Transformer Input
Self Annotation
```

extension SupervisedEstimator

```
/// Fits a transformer to an async sequence of examples.
///
/// Note that the async sequence is collected before fitting the estimator. This
may result in increased memory
/// use. Consider using the update method to train progressively in batches.
///
/// - Parameters:
///   - input: An async sequence of examples used for fitting the
transformer.
///   - eventHandler: An event handler.
/// - Returns: The fitted transformer.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func fitted Input Input
EventHandler async throws Self Transformer where Input
AsyncSequence Input Element
AnnotatedFeature Self Transformer Input Self Annotation

/// Fits a transformer to an async sequence of examples while validating with
a validation sequence.
///
/// Note that the async sequence is collected before fitting the estimator. This
may result in increased memory
/// use. Consider using the update method to train progressively in batches.
///
/// - Parameters:
///   - input: An async sequence of examples used for fitting the
transformer.
///   - validation: A sequence of examples used for validating the
fitted transformer.
```

```
/// - eventHandler: An event handler.
/// - Returns: The fitted transformer.
@available(macOS 14.0, iOS 17.0, tvOS 17.0, watchOS 11.0)

public func fitted<Input: Validation>() async throws Self<Transformer<Input, Validation>> where Input: AsyncSequence<Validation>, Sequence<Input>: Element, Input: AnnotatedFeature<Self, Transformer<Input, SelfAnnotation>>, Self: Transformer<Input, SelfAnnotation>, Validation: Element, Validation: AnnotatedFeature<Self, Transformer<Input, SelfAnnotation>>, SelfAnnotation: SelfAnnotation

/// A supervised temporal estimator wrapping a supervised estimator.
@available(iOS 13.0, tvOS 15.0)
@available(watchOS 6.0, tvOS 16.0, iOS 18.0)
@available(watchOS 7.0, tvOS 18.0)
@available(watchOS 8.0, iOS 1.0, tvOS 2.0)
@available(watchOS 9.0, iOS 16.0)
public struct SupervisedEstimatorToTemporalAdaptor<Base: SupervisedTemporalEstimator> where Base: SupervisedEstimator, Base.Annotation: Sendable

/// The transformer type created by this estimator.
public typealias Transformer = TransformerToTemporalAdaptor<Base>

/// The input type.
public typealias Input = Base.Transformer.Input

/// The output type.
public typealias Output = Base.Transformer.Output

/// The annotation type.
public typealias Annotation = Base.Annotation

/// Creates a temporal supervised estimator from a supervised estimator.
///
/// The resulting estimator collects all elements of the input sequence before
/// calling fit on the underlying
/// estimator. The transformer returned from fit is also converted to a
/// temporal transformer.
public init(_ base: Base)

/// Fits a transformer to a sequence of examples.
///
/// - Parameters:
///   - input: A sequence of examples used for fitting the transformer.
///   - eventHandler: An event handler.
/// - Returns: The fitted transformer.
```

```

@inlinable public func fitted InputSequence
FeatureSequence           InputSequence
EventHandler   nil async throws
SupervisedEstimatorToTemporalAdaptor Base Transformer where
InputSequence Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Base Annotation FeatureSequence Feature
Base Transformer Input

    /// Fits a transformer to a sequence of examples while validating with a
validation sequence.
    ///
    /// - Parameters:
    /// - input: A sequence of examples used for fitting the transformer.
    /// - validation: A sequence of examples used for validating the
fitted transformer.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
@inlinable public func fitted InputSequence Validation
FeatureSequence           InputSequence
Validation           EventHandler nil
async throws
SupervisedEstimatorToTemporalAdaptor Base Transformer where
InputSequence Sequence Validation Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Base Annotation
Validation Element AnnotatedFeature FeatureSequence
Base Annotation FeatureSequence Feature
Base Transformer Input

    /// Encodes a fitted transformer.
@inlinable public func encode _
SupervisedEstimatorToTemporalAdaptor Base Transformer
    inout any EstimatorEncoder throws

    /// Decodes the transformer.
@inlinable public func decode           inout any
EstimatorDecoder throws
SupervisedEstimatorToTemporalAdaptor Base Transformer

@available           13.0           15.0
@available           16.0           18.0
@available           16.0           18.0
@available           1.0            2.0
@available
extension SupervisedEstimatorToTemporalAdaptor Sendable
where Base Sendable

```

```
/// A tabular estimator that creates a transformer by fitting to a data set in a data
frame.
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public protocol SupervisedTabularEstimator

    /// The transformer type created by this estimator.
    associatedtype Transformer : TabularTransformer

    /// The annotation type.
    associatedtype Annotation

    /// The annotation column identifier.
    var annotationColumnID : ColumnID { Self.Annotation }

    /// Fits a transformer to a data frame
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples used for fitting the
    transformer.
    ///   - validation: A data frame containing examples used for
    validating the fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    func fitted(DataFrame) throws Self Transformer

    /// Encodes a fitted transformer.
    func encode(inout any EstimatorEncoder) throws Self Transformer

    /// Decodes a previously fitted transformer.
    func decode(inout any EstimatorDecoder) throws Self Transformer

extension SupervisedTabularEstimator

    @available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
    @inlinable public func fitted(DataFrame) throws Self Transformer

    @available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
    extension SupervisedTabularEstimator where Self : Encodable
```

```
/// Encodes a fitted encodable transformer.
public func encode _           Self Transformer
    inout any EstimatorEncoder throws

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SupervisedTabularEstimator where Self Transformer
Decodable

/// Decodes a previously fitted decodable transformer.
public func decode           inout any
EstimatorDecoder throws      Self Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension SupervisedTabularEstimator

/// Writes the encoded transformer to a file.
///
/// - Parameters:
///   - transformer: A transformer created by this estimator.
///   - url: A file URL.
///   - overwrite: A Boolean value indicating whether to overwrite
existing files.
public func write _           Self Transformer
URL           Bool true throws

/// Reads the encoded transformer from a file.
///
/// - Parameter url: A file URL.
/// - Returns: The decoded transformer.
public func read           URL throws      Self Transformer

extension SupervisedTabularEstimator

/// Composes this supervised tabular estimator with a tabular transformer.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func appending Other _           Other some
SupervisedTabularEstimator ComposedTabularTransformer Self Tra
nsformer Other Self Annotation where Other
TabularTransformer

/// Composes this tabular supervised estimator with a tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other _ Other some
SupervisedTabularEstimator ComposedTabularTransformer Self Transformer Other Transformer Self Annotation where Other TabularEstimator
```

```
/// Composes this supervised tabular estimator with another supervised tabular estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other _ Other some
SupervisedTabularEstimator ComposedTabularTransformer Self Transformer Other Transformer Self Annotation where Other SupervisedTabularEstimator Self Annotation Other Annotation
```

```
/// An estimator that creates a transformer by fitting to a sequence of annotated temporal features.
```

```
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public protocol SupervisedTemporalEstimator
```

```
/// The transformer type created by this estimator.
```

```
associatedtype Transformer : TemporalTransformer
```

```
/// The annotation type.
```

```
associatedtype Annotation : Equatable Sendable
```

```
/// Fits a transformer to a sequence of examples.
```

```
///
```

```
/// - Parameters:
```

```
///   - input: A sequence of annotated temporal sequences used for fitting the transformer.
```

```
///   - eventHandler: An event handler.
```

```
/// - Returns: The fitted transformer.
```

```
func fitted InputSequence FeatureSequence
```

```
InputSequence EventHandler async throws
Self Transformer where InputSequence : Sequence
FeatureSequence : TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Self Annotation
FeatureSequence Feature : Self Transformer Input
```

```
/// Fits a transformer to a sequence of examples while validating with a validation sequence.
```

```

    /**
     * - Parameters:
     *   - input: A sequence of annotated temporal sequences used for fitting the transformer.
     *   - validation: A sequence of examples used for validating the fitted transformer.
     *   - eventHandler: An event handler.
     * - Returns: The fitted transformer.
     func fitted InputSequence Validation FeatureSequence
           InputSequence Validation
           EventHandler async throws Self Transformer
where InputSequence Sequence Validation Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Self Annotation
Validation Element AnnotatedFeature FeatureSequence
Self Annotation FeatureSequence Feature
Self Transformer Input

    /// Encodes a fitted transformer.
    func encode _ Self Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    func decode inout any EstimatorDecoder
throws Self Transformer

```

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		
extension SupervisedTemporalEstimator		

```

    /// Writes the encoded transformer to a file.
    /**
     * - Parameters:
     *   - transformer: A transformer created by this estimator.
     *   - url: A file URL.
     *   - overwrite: A Boolean value indicating whether to overwrite existing files.
     public func write _ Self Transformer
URL Bool true throws

    /// Reads the encoded transformer from a file.
    /**
     * - Parameter url: A file URL.
     * - Returns: The decoded transformer.
     public func read URL throws Self Transformer

```

```

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension SupervisedTemporalEstimator

    /// Composes this supervised temporal estimator with a transformer.
    public func appending Other - Other some
    SupervisedTemporalEstimator ComposedTemporalTransformer Self T
    transformer TransformerToTemporalAdaptor Other
    Self Annotation where Other Transformer Other Input
    Self Transformer Output

    /// Composes this supervised temporal estimator with an estimator.
    public func appending Other - Other some
    SupervisedTemporalEstimator ComposedTemporalTransformer Self T
    transformer TransformerToTemporalAdaptor Other Transformer
    Self Annotation where Other Estimator
    Self Transformer Output Other Transformer Input

    /// Composes this supervised temporal estimator with a supervised
    estimator.
    public func appending Other - Other some
    SupervisedTemporalEstimator ComposedTemporalTransformer Self T
    transformer TransformerToTemporalAdaptor Other Transformer
    Self Annotation where Other SupervisedEstimator
    Self Annotation Other Annotation Self Transformer Output
    Other Transformer Input

    /// Composes this supervised temporal estimator with a transformer.
    public func appending Other - Other some
    SupervisedTemporalEstimator ComposedTemporalTransformer Self T
    transformer Other Self Annotation where Other
    TemporalTransformer Other Input Self Transformer Output

    /// Composes this supervised temporal estimator with a temporal estimator.
    public func appending Other - Other some
    SupervisedTemporalEstimator ComposedTemporalTransformer Self T
    transformer Other Transformer Self Annotation where Other
    TemporalEstimator Self Transformer Output
    Other Transformer Input

```

```

    /// Composes this supervised temporal estimator with another supervised
    temporal estimator.
    public func appending Other _ Other some
SupervisedTemporalEstimator ComposedTemporalTransformer Self T
ransformer Other Transformer Self Annotation where Other
SupervisedTemporalEstimator Self Annotation
Other Annotation Self Transformer Output
Other Transformer Input

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension SupervisedTemporalEstimator

    @inlinable public func fitted InputSequence
FeatureSequence InputSequence async throws
Self Transformer where InputSequence Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Self Annotation
FeatureSequence Feature Self Transformer Input

    @inlinable public func fitted InputSequence Validation
FeatureSequence InputSequence
Validation async throws Self Transformer where
InputSequence Sequence Validation Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Self Annotation
Validation Element AnnotatedFeature FeatureSequence
Self Annotation FeatureSequence Feature
Self Transformer Input

/// A tabular estimator that creates a transformer by fitting to a data set in a data
frame.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol TabularEstimator

    /// The transformer type created by this estimator.
    associatedtype Transformer TabularTransformer

    /// Fits a transformer to a data frame
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.

```

```
func fitted DataFrame
EventHandler  async throws Self Transformer

    /// Encodes a fitted transformer.
    func encode _ Self Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    func decode inout any EstimatorDecoder
throws Self Transformer

extension TabularEstimator

    /// Exposes this tabular estimator as a supervised tabular estimator.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

        public func
adaptedAsSupervised Annotation
ColumnID Annotation
TabularEstimatorToSupervisedAdaptor Self Annotation

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TabularEstimator

    /// Writes the encoded transformer to a file.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - url: A file URL.
    ///   - overwrite: A Boolean value indicating whether to overwrite
existing files.
    public func write _ Self Transformer
URL Bool true throws

    /// Reads the encoded transformer from a file.
    ///
    /// - Parameter url: A file URL.
    /// - Returns: The decoded transformer.
    public func read URL throws Self Transformer

extension TabularEstimator

    /// Compose this tabular estimator with a tabular transformer.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

        public func appending Other _ Other some
```

```
TabularEstimator ComposedTabularTransformer Self Transformer
Other where Other TabularTransformer
```

```
/// Compose this tabular estimator with another tabular estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other _ Other some
```

```
TabularEstimator ComposedTabularTransformer Self Transformer
Other Transformer where Other TabularEstimator
```

```
extension TabularEstimator
```

```
/// Composes this tabular estimator with a supervised tabular estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public func appending Other _ Other some
```

```
SupervisedTabularEstimator ComposedTabularTransformer Self Tra
nsformer Other Transformer Other Annotation where Other
SupervisedTabularEstimator
```

```
extension TabularEstimator where Self Transformer Encodable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
@inlinable public func fitted DataFrame async
throws Self Transformer
```

```
/// Encodes a fitted encodable transformer.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
@inlinable public func encode inout any EstimatorEncoder
Self Transformer throws
```

```
extension TabularEstimator
```

```
@available macOS 15.0 iOS 18.0 tvOS 18.0 watchOS 11.0
```

```
@inlinable public func fitted DataFrame async
throws Self Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TabularEstimator where Self Transformer Decodable
```

```
    /// Decodes a previously fitted decodable transformer.
    @inlinable public func decode inout any
EstimatorDecoder throws Self Transformer
```

```
    /// An adaptor that exposes a tabular estimator as a tabular supervised estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct TabularEstimatorToSupervisedAdaptor Estimator
Annotation SupervisedTabularEstimator where Estimator
TabularEstimator
```

```
    /// The transformer type created by this estimator.
public typealias Transformer Estimator Transformer
```

```
    /// The annotation column identifier.
public var annotationColumnID ColumnID Annotation
```

```
    /// The wrapped estimator.
public let estimator Estimator
```

```
    /// Creates a tabular estimator supervised adaptor.
public init _ Estimator
ColumnID Annotation
```

```
    /// Returns the tabular transformer fitted using the provided tabular estimator.
    ///
```

```
    /// - Parameters:
```

```
    /// - input: A data frame containing examples.
    /// - validation: A data frame containing examples.
    /// - eventHandler: An event handler.
```

```
    /// - Returns: The a fitted tabular transformer.
```

```
    @inlinable public func fitted DataFrame
        DataFrame nil
EventHandler nil async throws Estimator Transformer
```

```
    /// Encodes a fitted transformer.
```

```
    public func encode _ Estimator Transformer
        inout any EstimatorEncoder throws
```

```
    /// Decodes a previously fitted transformer.
```

```
    public func decode inout any
EstimatorDecoder throws Estimator Transformer
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```

extension TabularEstimatorToSupervisedAdaptor Sendable where
Estimator Sendable Annotation Sendable

/// Errors related to tabular pipeline data affinity problems.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public enum TabularPipelineDataError LocalizedError
Equatable Sendable

    /// A column is missing from the data frame.
    case missingColumn String String

    /// A column has an incorrect type.
    case incorrectType String String
        String String

    /// The selected column has missing values.
    case missingValues String String

    /// A localized message describing what error occurred.
    public var errorDescription String get

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
    public static func TabularPipelineDataError
TabularPipelineDataError Bool

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TabularPipelineDataError
CustomDebugStringConvertible

    /// A text representation of the error.
    public var debugDescription String get

    /// A tabular transformer that transforms a data frame.
    ///
    /// Tabular transformers represent operations on data frames. They modify and
    /// operate on values on one or more columns.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol TabularTransformer Transformer where
Self Input DataFrame Self Output DataFrame

```

```
extension TabularTransformer

    /// Composes this transformer with an updatable estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func appending Other _ Other
    PreprocessingUpdatableTabularEstimator Self Other where
    Other UpdatableTabularEstimator


extension TabularTransformer

    /// Exports this transformer as a CoreML model.
    ///
    /// – Note: By default this method exports .mlpackage files. You can
    export a .mlmodel file by specifying that as
    /// the URL file extension. But if you specify .mlmodel and the transformer
    doesn't support it, this method will
    /// throw an error.
    ///
    /// – Parameter url: The location to write the model into.
    @available macOS 13.0 iOS 16.0 tvOS 16.0
    public func export URL throws

    /// Exports this tabular transformer as a CoreML model with userInfo.
    ///
    /// – Note: By default this method exports .mlpackage files. You can
    export a .mlmodel file by specifying that as
    /// the URL file extension. But if you specify .mlmodel and the transformer
    doesn't support it, this method will
    /// throw an error.
    ///
    /// – Parameters:
    /// – url: The location to write the model into.
    /// – metadata: Contextual user-provided information.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public func export URL ModelMetadata
throws


extension TabularTransformer

    /// Composes this transformer with an updatable supervised estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func appending Other _ Other
    PreprocessingUpdatableSupervisedTabularEstimator Self Other where
    Other UpdatableSupervisedTabularEstimator
```

```
extension TabularTransformer

    /// Composes this tabular transformer with another tabular transformer.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other
ComposedTabularTransformer Self Other where Other
TabularTransformer

extension TabularTransformer

    /// Compose this tabular transformer with a tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other some
TabularEstimator ComposedTabularTransformer Self
Other Transformer where Other TabularEstimator

extension TabularTransformer

    /// Composes this transformer with a supervised tabular estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func appending Other _ Other
PreprocessingSupervisedTabularEstimator Self Other where
Other SupervisedTabularEstimator

extension TabularTransformer

    /// Exposes this tabular transformer as a trivial tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    @inlinable public func adaptedAsEstimator
TabularTransformerToEstimatorAdaptor Self

extension TabularTransformer

    /// Exposes this tabular transformer as an updatable tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    @inlinable public func adaptedAsUpdatableEstimator
TabularTransformerToUpdateableEstimatorAdaptor Self
```

```
extension TabularTransformer

    /// Composes this transformer with a supervised tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other some
SupervisedTabularEstimator ComposedTabularTransformer Self
Other Transformer Other Annotation where Other
SupervisedTabularEstimator

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TabularTransformer

    /// Performs the transformation on a single input.
///
/// - Parameters:
///   - input: The transformer input.
///   - eventHandler: An event handler.
/// - Returns: An output produced by applying the transformer to the

input.
@inlinable public func callAsFunction _
EventHandler nil async throws DataFrame
DataFrame

extension TabularTransformer

    /// Composes this transformer with an estimator.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func appending Other _ Other
PreprocessingTabularEstimator Self Other where Other
TabularEstimator

/// A tabular estimator that always returns a predefined tabular transformer.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct
TabularTransformerToEstimatorAdaptor Transformer
TabularEstimator where Transformer TabularTransformer

/// A pre-defined tabular transformer.
public let transformer Transformer

/// Creates a trivial tabular estimator.
public init _ Transformer
```

```

    /// Returns the pre-defined tabular transformer.
    ///
    /// - Parameters:
    ///   - input: A data frame.
    ///   - eventHandler: An event handler.
    /// - Returns: The pre-defined tabular transformer.
    @inlinable public func fitted           DataFrame
        EventHandler    nil  async throws   Transformer

    /// Does nothing since this tabular estimator uses a pre-defined tabular
transformer.
    @inlinable public func encode _           Transformer
        inout any EstimatorEncoder throws

    /// Returns the pre-defined tabular transformer.
    @inlinable public func decode           inout any
EstimatorDecoder  throws   Transformer

@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
extension TabularTransformerToEstimatorAdaptor  Sendable
where Transformer  Sendable

    /// An updatable tabular estimator that always returns a predefined transformer.
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public struct
TabularTransformerToUpdateableEstimatorAdaptor Transformer
UpdatableTabularEstimator where Transformer
TabularTransformer

    /// A pre-defined transformer.
    public let transformer  Transformer

    /// Creates an updatable tabular estimator from a tabular transformer.
    public init _           Transformer

    /// Creates a default-initialized transformer suitable for incremental fitting.
    @inlinable public func makeTransformer   Transformer

    /// Returns the pre-defined transformer.
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The pre-defined transformer.
    @inlinable public func fitted           DataFrame
        EventHandler    nil  async throws   Transformer

```

```
    /// Does nothing since this estimator uses a pre-defined transformer.
    @inlinable public func update _ inout
Transformer           DataFrame
EventHandler      nil  async throws

    /// Does nothing since this estimator uses a pre-defined transformer.
    @inlinable public func encode _ Transformer
        inout any EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    @inlinable public func decode inout any
EstimatorDecoder throws Transformer

    /// Does nothing since this estimator uses a pre-defined transformer.
    @inlinable public func encodeWithOptimizer _
Transformer           inout any EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    @inlinable public func decodeWithOptimizer
inout any EstimatorDecoder throws Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TabularTransformerToUpdatableEstimatorAdaptor
Sendable where Transformer Sendable

/// A temporal transformer that applies a regular transformer to each value of a
temporal sequence.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct TemporalAdaptor Base TemporalTransformer
Sendable where Base Transformer Base Sendable

    /// The input type.
    public typealias Input Base Input

    /// The output type.
    public typealias Output Base Output

    /// The output sequence type.
    public typealias OutputSequence
AnyTemporalSequence TemporalAdaptor Base Output

    /// Creates a temporal transformer from a transformer.
    ///
    /// The resulting transformer applies the underlying transformer to each
element in the input sequence.
    public init _ Base
```

```

    /// Performs the transformation on each element of the input sequence.
    public func applied some
TemporalSequence Base Input EventHandler
nil async throws
AnyTemporalSequence TemporalAdaptor Base Output

    /// An estimator that creates a transformer by fitting to a sequence of temporal
features.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public protocol TemporalEstimator

    /// The transformer type created by this estimator.
associatedtype Transformer TemporalTransformer

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    func fitted InputSequence InputSequence
EventHandler async throws Self Transformer
where InputSequence Sequence InputSequence Element
TemporalSequence Self Transformer Input
InputSequence Element Feature

    /// Encodes a fitted transformer.
    func encode _ Self Transformer
inout any EstimatorEncoder throws

    /// Decodes a previously fitted transformer.
    func decode inout any EstimatorDecoder
throws Self Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension TemporalEstimator

    /// Writes the encoded transformer to a file.

```

```

    /**
     * - Parameters:
     *   - transformer: A transformer created by this estimator.
     *   - url: A file URL.
     *   - overwrite: A Boolean value indicating whether to overwrite
existing files.
    public func write _ Self Transformer
URL Bool _true throws

    /// Reads the encoded transformer from a file.
    /**
     * - Parameter url: A file URL.
     * - Returns: The decoded transformer.
    public func read _ URL throws Self Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension TemporalEstimator

    /// Exposes this temporal estimator as a supervised temporal estimator.
    public func
adaptedAsSupervised Annotation
Annotation self
TemporalEstimatorToSupervisedAdaptor Self Annotation where
Annotation Equatable Annotation Sendable

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension TemporalEstimator

    @inlinable public func fitted InputSequence
InputSequence async throws Self Transformer where
InputSequence Sequence InputSequence Element
TemporalSequence Self Transformer Input
InputSequence Element Feature

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0

```

```
@available  
extension TemporalEstimator
```

```
    /// Composes this temporal estimator with a supervised temporal estimator.
```

```
    @preconcURRENCY public func appending Other _
```

```
Other some
```

```
SupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer  
Transformer TransformerToTemporalAdaptor Other Transformer
```

```
Other Annotation where Other SupervisedEstimator
```

```
Other Annotation Sendable Self Transformer Output
```

```
Other Transformer Input
```

```
    /// Composes this temporal estimator with a supervised temporal estimator.
```

```
    public func appending Other _ Other some
```

```
SupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer  
Other Transformer Other Annotation where
```

```
Other SupervisedTemporalEstimator Self Transformer Output
```

```
Other Transformer Input
```

```
@available
```

```
13.0
```

```
15.0
```

```
@available
```

```
16.0
```

```
18.0
```

```
@available
```

```
16.0
```

```
18.0
```

```
@available
```

```
1.0
```

```
2.0
```

```
@available
```

```
extension TemporalEstimator
```

```
    /// Composes this temporal estimator with a transformer.
```

```
    public func appending Other _ Other some
```

```
TemporalEstimator ComposedTemporalTransformer Self Transformer
```

```
TransformerToTemporalAdaptor Other where Other
```

```
Transformer Other Input Self Transformer Output
```

```
    /// Composes this temporal estimator with an estimator.
```

```
    public func appending Other _ Other some
```

```
TemporalEstimator ComposedTemporalTransformer Self Transformer
```

```
TransformerToTemporalAdaptor Other Transformer where
```

```
Other Estimator Self Transformer Output
```

```
Other Transformer Input
```

```
    /// Composes this temporal estimator with a temporal transformer.
```

```
    public func appending Other _ Other some
```

```
TemporalEstimator ComposedTemporalTransformer Self Transformer
```

```
Other where Other TemporalTransformer Other Input
```

```
Self Transformer Output
```

```

    /// Composes this temporal estimator with another temporal estimator.
    public func appending Other _           Other      some
TemporalEstimator ComposedTemporalTransformer Self Transformer
    Other Transformer  where Other TemporalEstimator
Self Transformer Output  Other Transformer Input

    /// An adaptor that exposes a temporal estimator as a supervised temporal
estimator.
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
public struct TemporalEstimatorToSupervisedAdaptor Estimator
Annotation SupervisedTemporalEstimator where Estimator
TemporalEstimator Annotation Equatable Annotation
Sendable

    /// The transformer type created by this estimator.
public typealias Transformer Estimator Transformer

    /// The wrapped estimator.
public let estimator Estimator

    /// Creates a temporal estimator adaptor.
public init _

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted InputSequence FeatureSequence
        InputSequence           EventHandler nil async
throws TemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where InputSequence Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Annotation
FeatureSequence Feature Estimator Transformer Input

    /// Fits a transformer to a sequence of examples while validating with a
validation sequence.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.

```

```
    /// - validation: A sequence of examples used for validating the
    fitted transformer.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted InputSequence Validation
FeatureSequence           InputSequence
                           Validation           EventHandler     nil
async throws
TemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where InputSequence Sequence
Validation Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Annotation Validation Element
AnnotatedFeature FeatureSequence Annotation
FeatureSequence Feature Estimator Transformer Input
```

```
    /// Does nothing since this estimator uses a pre-defined transformer.
```

```
    public func encode _
TemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer           inout any
EstimatorEncoder throws
```

```
    /// Returns the pre-defined transformer.
```

```
    public func decode           inout any
EstimatorDecoder throws
TemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer
```

```
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
extension TemporalEstimatorToSupervisedAdaptor Sendable
where Estimator Sendable
```

```
    /// A temporal feature contains a segment identifier and a feature value.
```

```
available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct TemporalFeature Feature Identifiable
```

```
    /// The temporal segment identifier for this feature.
```

```
    public var id TemporalSegmentIdentifier
```

```
    /// The feature value.
```

```
    public var feature Feature
```

```
    /// Creates a temporal feature.
```

```
///  
/// - Parameters:  
///   - id: The temporal segment identifier.  
///   - feature: The feature value.  
public init      TemporalSegmentIdentifier  
Feature  
  
/// A type representing the stable identity of the entity associated with  
/// an instance.  
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
public typealias ID    TemporalSegmentIdentifier  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFeature Equatable where Feature Equatable  
  
/// Returns a Boolean value indicating whether two values are equal.  
///  
/// Equality is the inverse of inequality. For any values `a` and `b`,  
/// `a == b` implies that `a != b` is `false`.  
///  
/// - Parameters:  
///   - lhs: A value to compare.  
///   - rhs: Another value to compare.  
public static func      TemporalFeature Feature  
TemporalFeature Feature Bool  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFeature Hashable where Feature Hashable  
  
/// Hashes the essential components of this value by feeding them into the  
/// given hasher.  
///  
/// Implement this method to conform to the `Hashable` protocol. The  
/// components used for hashing must be the same as the components  
compared  
/// in your type's `==` operator implementation. Call  
`hasher.combine(_:)`  
/// with each of these components.  
///  
/// - Important: In your implementation of `hash(into:)`,  
/// don't call `finalize()` on the `hasher` instance provided,  
/// or replace it with a different instance.  
/// Doing so may become a compile-time error in the future.  
///
```

```
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TemporalFeature Encodable where Feature
Encodable
```

```
    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TemporalFeature Decodable where Feature
Decodable
```

```
    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or
    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TemporalFeature Sendable where Feature Sendable
```

```
/// A URL and a time range identifying a specific segment of a time-based  
(temporal) file.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public struct TemporalFileSegment
```

```
/// The file URL.  
public var url URL
```

```
/// The time range in seconds.  
public var range Range TimeInterval
```

```
/// Creates a TemporalFileSegment.
```

```
///
```

```
/// - Parameters:
```

```
/// - url: A file URL.
```

```
/// - range: A time range in seconds.
```

```
public init URL Range TimeInterval
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFileSegment Sendable
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFileSegment Equatable
```

```
/// Returns a Boolean value indicating whether two values are equal.
```

```
///
```

```
/// Equality is the inverse of inequality. For any values `a` and `b`,  
/// `a == b` implies that `a != b` is `false`.
```

```
///
```

```
/// - Parameters:
```

```
/// - lhs: A value to compare.
```

```
/// - rhs: Another value to compare.
```

```
public static func TemporalFileSegment
```

```
TemporalFileSegment Bool
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFileSegment Hashable
```

```
/// Hashes the essential components of this value by feeding them into the  
/// given hasher.
```

```
///
```

```
/// Implement this method to conform to the `Hashable` protocol. The  
/// components used for hashing must be the same as the components  
compared
```

```
/// in your type's `==` operator implementation. Call
```

```
`hasher.combine(_:)`  
    /// with each of these components.  
    ///  
    /// - Important: In your implementation of `hash(into:)`,  
    /// don't call `finalize()` on the `hasher` instance provided,  
    /// or replace it with a different instance.  
    /// Doing so may become a compile-time error in the future.  
    ///  
    /// - Parameter hasher: The hasher to use when combining the  
components  
    /// of this instance.  
public func hash           inout Hasher  
  
    /// The hash value.  
    ///  
    /// Hash values are not guaranteed to be equal across different executions of  
    /// your program. Do not save hash values to use during a future execution.  
    ///  
    /// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To  
    /// conform to `Hashable`, implement the `hash(into:)` requirement  
instead.  
    /// The compiler provides an implementation for `hashValue` for you.  
public var hashValue Int get
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFileSegment Encodable  
  
    /// Encodes this value into the given encoder.  
    ///  
    /// If the value fails to encode anything, `encoder` will encode an empty  
    /// keyed container in its place.  
    ///  
    /// This function throws an error if any values are invalid for the given  
    /// encoder's format.  
    ///  
    /// - Parameter encoder: The encoder to write data to.  
public func encode           any Encoder throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalFileSegment Decodable  
  
    /// Creates a new instance by decoding from the given decoder.  
    ///  
    /// This initializer throws an error if reading from the decoder fails, or  
    /// if the data read is corrupted or otherwise invalid.  
    ///  
    /// - Parameter decoder: The decoder to read data from.
```

```
public init           any Decoder throws

/// Uniquely identifies a segment of a temporal sequence.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public struct TemporalSegmentIdentifier : Hashable, Sendable

/// The segment source. For files use the full path or URL of the file.
public var source : String

/// The segment's timestamp range.
///
/// To get a timestamp in seconds divide the value by the timescale.
public var range : Range<Int>

/// The identifier's timescale is the number of uniquely identifiable
timestamps in a second.
///
/// For example an audio file sampled at 44,100 Hz should have a timescale
value of 44,100 (or an integer multiple
/// of that) so that every sample has a unique timestamp.
public var timescale : Int

/// The time range in seconds.
public var rangeInSeconds : Range<TimeInterval> get

/// The segment duration in seconds.
public var durationInSeconds : TimeInterval get

/// Creates a temporal-segment identifier.
///
/// - Parameters:
///   - source: A unique source description.
///   - range: A timestamp range.
///   - timescale: The number of uniquely identifiable timestamps in a
second.
public init(source: String, range: Range<Int>, timescale: Int)

/// Hashes the essential components of this value by feeding them into the
/// given hasher.
///
/// Implement this method to conform to the `Hashable` protocol. The
/// components used for hashing must be the same as the components
compared
/// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
/// with each of these components.
///
/// - Important: In your implementation of `hash(into:)`,
```

```

    //// don't call `finalize()` on the `hasher` instance provided,
    //// or replace it with a different instance.
    //// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining the
components
    /// of this instance.
public func hash inout Hasher

    /// Returns a Boolean value indicating whether two values are equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and `b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func TemporalSegmentIdentifier
TemporalSegmentIdentifier Bool

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different executions of
    /// your program. Do not save hash values to use during a future execution.
    ///
    /// - Important: `hashValue` is deprecated as a `Hashable`
requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement
instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue Int get

```

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TemporalSegmentIdentifier Codable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

    /// Creates a new instance by decoding from the given decoder.
    ///
    /// This initializer throws an error if reading from the decoder fails, or

```

```

    /// if the data read is corrupted or otherwise invalid.
    ///
    /// - Parameter decoder: The decoder to read data from.
public init any Decoder throws

    /// Async sequence for temporal features.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol TemporalSequence AsyncSequence

    /// The feature type.
    associatedtype Feature where Self Element
TemporalFeature Self Feature

    /// The number of elements in the sequence if available, calculated
nondestructively.
var count Int get

    /// A transformer that takes an asynchronous input sequence of temporal features
and produces an asynchronous output sequence.
///
/// A temporal transformer, unlike a regular transformer, can accumulate multiple
inputs before producing an output.
/// For example, an audio transformer can accumulate audio buffers until the
desired length is reached before producing
/// an output.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol TemporalTransformer

    /// The input type.
    associatedtype Input

    /// The output type.
    associatedtype Output where Self Output
Self OutputSequence Feature

    /// The output async sequence type.
    associatedtype OutputSequence TemporalSequence

    /// Performs the transformation on an input sequence.
///
/// - Parameters:
///   - input: The input temporal sequence.
///   - eventHandler: An event handler.
/// - Returns: An async sequence produced by applying the
transformation to the input.
func applied S S EventHandler
async throws Self OutputSequence where S
TemporalSequence Self Input S Feature

```

```
extension TemporalTransformer
```

```
    /// Composes this temporal transformer with a transformer.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available  
        public func appending Other - Other  
ComposedTemporalTransformer Self -  
TransformerToTemporalAdaptor Other where Other  
Transformer Self Output Other Input
```

```
    /// Composes this temporal transformer with a transformer.
```

```
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
        public func appending Other - Other  
ComposedTemporalTransformer Self - TemporalAdaptor Other  
where Other Transformer Other Sendable Self Output  
Other Input
```

```
    /// Composes this temporal transformer with another temporal transformer.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
  
        public func appending Other - Other  
ComposedTemporalTransformer Self - Other where Other  
TemporalTransformer Self Output Other Input
```

```
extension TemporalTransformer
```

```
    /// Composes this transformer with an updatable supervised temporal estimator.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available  
        public func appending Other - Other  
PreprocessingUpdatableSupervisedTemporalEstimator Self -  
UpdatableSupervisedEstimatorToTemporalAdaptor Other where  
Other UpdatableSupervisedEstimator Self Output  
Other Transformer Input Other Annotation Sendable
```

```
    /// Composes this transformer with an updatable supervised temporal estimator.
```

```
    @available 13.0 15.0  
    @available 16.0 18.0
```

```
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingUpdatableSupervisedTemporalEstimator Self Other
where Other UpdatableSupervisedTemporalEstimator
Self Output Other Transformer Input
```

extension TemporalTransformer

```
/// Composes this temporal transformer with an updatable estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingUpdatableTemporalEstimator Self
UpdatableEstimatorToTemporalAdaptor Other where Other
UpdatableEstimator Self Output Other Transformer Input
```

```
/// Composes this temporal transformer with an updatable temporal
estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingUpdatableTemporalEstimator Self Other where
Other UpdatableTemporalEstimator Self Output
Other Transformer Input
```

extension TemporalTransformer

```
/// Composes this transformer with a supervised temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingSupervisedTemporalEstimator Self
SupervisedEstimatorToTemporalAdaptor Other where Other
SupervisedEstimator Self Output Other Transformer Input
Other Annotation Sendable
```

```
    /// Composes this transformer with a supervised temporal estimator.  
    @available(macOS 13.0, iOS 16.0, tvOS 16.0)  
    @available(watchOS 11.0)  
    @available(Other 1.0)  
    @available(Self 2.0)  
  
    public func appending<Other>(_ other: Other) -> PreprocessingSupervisedTemporalEstimator<Self, Other> where  
        Other: SupervisedTemporalEstimator<Self, Output>  
        Other: Transformer<Input>
```

extension TemporalTransformer

```
    /// Exports this temporal transformer as a CoreML model.  
    ///  
    /// - Note: By default this method exports .mlpackage files. You can  
    export a .mlmodel file by specifying that as  
        /// the URL file extension. But if you specify .mlmodel and the transformer  
        doesn't support it, this method will  
            /// throw an error.  
    ///  
    /// - Parameter url: The location to write the model into.  
    @available(macOS 13.0, iOS 16.0, tvOS 16.0)  
    public func export(url: URL) throws  
  
    /// Exports this temporal transformer as a CoreML model with user-supplied  
    metadata.  
    ///  
    /// - Note: By default this method exports .mlpackage files. You can  
    export a .mlmodel file by specifying that as  
        /// the URL file extension. But if you specify .mlmodel and the transformer  
        doesn't support it, this method will  
            /// throw an error.  
    ///  
    /// - Parameters:  
    ///     - url: The location to write the model into.  
    ///     - metadata: Contextual user-provided information.  
    @available(macOS 14.0, iOS 17.0, tvOS 17.0)  
    public func export(url: URL, metadata: ModelMetadata) throws
```

```
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)  
extension TemporalTransformer
```

```
    /// Performs the transformation on a sequence of input sequences.  
    ///  
    /// - Parameters:  
    ///     - input: The transformer inputs.
```

```
    /// - eventHandler: An event handler.  
    /// - Returns: The outputs produced by applying the transformer to the  
inputs.
```

```
    @inlinable public func applied S S  
        EventHandler nil async throws  
    Self OutputSequence where S Sequence Self Input  
S Element Feature S Element TemporalSequence
```

```
    /// Performs the transformation on a sequence of annotated input  
sequences.
```

```
    ///  
    /// - Parameters:  
    /// - input: A sequence of annotated sequences.  
    /// - eventHandler: An event handler.  
    /// - Returns: The annotated outputs produced by applying the  
transformer to the inputs.
```

```
    @inlinable public func applied S TS Annotation  
        S EventHandler nil async throws  
    AnnotatedFeature Self OutputSequence Annotation where S  
Sequence TS TemporalSequence Self Input TS Feature  
S Element AnnotatedFeature TS Annotation
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TemporalTransformer
```

```
    /// Performs the transformation on an input sequence.  
    ///  
    /// - Parameters:  
    /// - input: The input temporal sequence.  
    /// - eventHandler: An event handler.  
    /// - Returns: An async sequence produced by applying the  
transformation to the input.
```

```
    @inlinable public func callAsFunction S S  
        EventHandler nil async throws  
    Self OutputSequence where S TemporalSequence Self Input  
S Feature
```

```
    /// Performs the transformation on a sequence of inputs.  
    ///  
    /// - Parameters:  
    /// - input: The transformer inputs.  
    /// - eventHandler: An event handler.  
    /// - Returns: The outputs produced by applying the transformer to the  
inputs.
```

```
    @inlinable public func callAsFunction S S  
        EventHandler nil async throws  
    Self OutputSequence where S Sequence Self Input  
S Element Feature S Element TemporalSequence
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TemporalTransformer
```

```
    /// Performs a prediction on a single input.
    ///
    /// - Parameter input: The input feature.
    /// - Returns: A classification array.
    public func prediction S Label S async
throws Self OutputSequence where S TemporalSequence
Label Hashable Self Input S Feature Self Output
ClassificationDistribution Label
```

```
extension TemporalTransformer
```

```
    /// Exposes this temporal transformer as a trivial temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
@inlinable public func adaptedAsEstimator
TemporalTransformerToEstimatorAdaptor Self
```

```
extension TemporalTransformer
```

```
    /// Exposes this temporal transformer as a trivial temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
@inlinable public func adaptedAsUpdatableEstimator
TemporalTransformerToUpdateableEstimatorAdaptor Self
```

```
extension TemporalTransformer
```

```
    /// Composes this temporal transformer with an estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingTemporalEstimator Self
EstimatorToUpdateableEstimator Adaptor Other where Other Estimator
```

Self Output Other Transformer Input

/// Composes this temporal transformer with a temporal estimator.

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		

public func appending Other - Other

PreprocessingTemporalEstimator Self Other where Other
TemporalEstimator Self Output Other Transformer Input

/// A temporal estimator that always returns a predefined temporal transformer.

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		

public struct

TemporalTransformerToEstimatorAdaptor Transformer

TemporalEstimator where Transformer TemporalTransformer

/// A pre-defined transformer.

public let transformer Transformer

/// Creates a trivial estimator.

public init _ Transformer

/// Returns the pre-defined transformer.

///

/// - Parameters:

/// - input: A sequence of examples.

/// - eventHandler: An event handler.

/// - Returns: The fitted transformer.

@inlinable public func fitted InputSequence

InputSequence	EventHandler	nil	async	throws
Transformer	where	InputSequence	Sequence	
Transformer Input		InputSequence	Element	Feature
InputSequence Element		TemporalSequence		

/// Does nothing since this estimator uses a pre-defined transformer.

@inlinable public func encode _ Transformer
inout any EstimatorEncoder throws

/// Returns the pre-defined transformer.

@inlinable public func decode inout any

EstimatorDecoder throws Transformer

```

@available(13.0, 15.0)
@available(16.0, 18.0)
@available(16.0, 18.0)
@available(1.0, 2.0)
@available
extension TemporalTransformerToEstimatorAdaptor where Transformer : Sendable
/// A temporal estimator that always returns a predefined temporal transformer.
@available(13.0, 15.0)
@available(16.0, 18.0)
@available(16.0, 18.0)
@available(1.0, 2.0)
@available
public struct TemporalTransformerToUpdatableEstimatorAdaptor<T> where T : Transformer
TemporalTransformer<T>
/// A pre-defined transformer.
public let transformer : Transformer
/// Creates a trivial estimator.
public init _ : Transformer
/// Creates a default-initialized transformer suitable for incremental fitting.
@inlinable public func makeTransformer() : Transformer
/// Returns the pre-defined transformer.
///
/// - Parameters:
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
/// - Returns: The fitted transformer.
@inlinable public func fitted<InputSequence: InputSequence<Input, Element, Feature>, EventHandler: EventHandler<Input, Element, TemporalSequence>>(input: InputSequence, eventHandler: EventHandler) : nil async throws Transformer<Input, Sequence<Element, Feature>> where Input: InputSequence, Sequence<Element, Feature>: InputSequence, Element: TemporalSequence
/// Does nothing since this estimator uses a pre-defined transformer.
@inlinable public func update<InputSequence: InputSequence<Input, Element, Feature>, EventHandler: EventHandler<Input, Element, TemporalSequence>>(transformer: inout Transformer<Input, Element, TemporalSequence>, eventHandler: EventHandler) : nil async throws where InputSequence: Sequence<Transformer<Input, Element, TemporalSequence>, Input: InputSequence, Element: Feature, InputSequence: InputSequence<Element, Feature>>

```

```

/// Does nothing since this estimator uses a pre-defined transformer.
@inlinable public func encode _ Transformer
    inout any EstimatorEncoder throws

/// Returns the pre-defined transformer.
@inlinable public func decode inout any
EstimatorDecoder throws Transformer

/// This method is part of the conformance. It doesn't encode anything since
the transformer is pre-defined, so don't call it.
@inlinable public func encodeWithOptimizer _ Transformer
    inout any EstimatorEncoder throws

/// Returns the pre-defined transformer.
@inlinable public func decodeWithOptimizer
inout any EstimatorDecoder throws Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension TemporalTransformerToUpdatableEstimatorAdaptor
Sendable where Transformer Sendable

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
@available
public struct TimeSeriesClassifier Scalar Label
SupervisedEstimator where Scalar MLShapedArrayScalar Scalar
    BinaryFloatingPoint Label Comparable Label Decodable
Label Encodable Label Hashable

/// The transformer type created by this estimator.
public typealias Transformer
TimeSeriesClassifier Scalar Label Model

/// The annotation type.
public typealias Annotation Label

/// The configuration.
public var configuration
TimeSeriesClassifierConfiguration

/// The set of possible labels.
public var labels Set Label

```

```

    /// Creates a time series classifier.
    ///
    /// - Parameters:
    ///   - labels: The labels used to train the classifier.
    ///   - configuration: The configuration.
    public init Set Label
TimeSeriesClassifierConfiguration

    /// Fits a time series classifier model to a sequence of examples.
    ///
    /// The training process partitions the input into random batches according to
    the batch size configuration
    /// parameter. Training stops when the validation loss stops improving or
    when the maximum number of iterations
    /// is reached.
    ///
    /// - Parameters:
    ///   - input: A sequence of annotated features for training. Each
    feature's shape should be `[sequenceLength, featureSize]`
    /// and each annotation should be one-hot encoded with shape
    `[labelCount]`.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted time series classifier model.
    public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar Label
EventHandler nil async throws
TimeSeriesClassifier Scalar Label Model

    /// Fits a time series classifier model to a sequence of examples.
    ///
    /// The training process partitions the input into random batches according to
    the batch size configuration
    /// parameter. Training stops when the validation loss stops improving or
    when the maximum number of iterations
    /// is reached.
    ///
    /// - Parameters:
    ///   - input: A sequence of annotated features for training. Each
    feature's shape should be `[sequenceLength, featureSize]`.
    ///   - validation: A sequence of annotated features for validating.
    Each feature's shape should be `[sequenceLength, featureSize]`.
    /// - eventHandler: An event handler.
    /// - Returns: The fitted time series classifier model.
    public func fitted some
Sequence AnnotatedFeature MLShapedArray Scalar Label
some
Sequence AnnotatedFeature MLShapedArray Scalar Label
EventHandler nil async throws
TimeSeriesClassifier Scalar Label Model

```

```
@available
extension TimeSeriesClassifier

    /// A time-series classifier model.
    ///
    /// - Note: Only `Float` and `Double` are currently supported as the
Scalar type.
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0

@available
public struct Model : Transformer : @unchecked Sendable

    /// The input type.
    public typealias Input = MLShapedArray

    /// The output type.
    public typealias Output =
ClassificationDistribution Label

    /// The number of samples between temporal predictions.
public var stride : Int

    /// Performs a classification on a shaped array of input features.
    ///
    /// - Parameters:
    ///   - input: A shaped array of input features. The shape must
`[sequenceLength, featureSize]`.
    ///   - eventHandler: An event handler.
    /// - Returns: A classification distribution.
    public func applied : MLShapedArray
        EventHandler nil async throws
ClassificationDistribution Label

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
@available
extension TimeSeriesClassifier : UpdatableSupervisedEstimator

    /// Creates a default-initialized model suitable for incremental fitting.
    public func makeTransformer
TimeSeriesClassifier Scalar Label Model

    /// Updates a model with a new batch of examples.
    ///
    /// - Parameters:
```

```

    /// - transformer: A transformer to update.
    /// - input: A sequence of annotated features for updating the
transformer. Each feature's shape should be
    /// `[sequenceLength, featureSize]`.
    /// - eventHandler: An event handler.
public func update inout
TimeSeriesClassifier Scalar Label Model some
Sequence AnnotatedFeature MLShapedArray Scalar Label
EventHandler nil async throws

    /// Encodes the model and optimizer to an encoder.
public func encodeWithOptimizer inout
TimeSeriesClassifier Scalar Label Transformer
inout any EstimatorEncoder throws

    /// Reads the encoded model and optimizer with a decoder.
public func decodeWithOptimizer inout any
EstimatorDecoder throws TimeSeriesClassifier Scalar
Label Transformer

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
@available
extension TimeSeriesClassifier

    public typealias Configuration
TimeSeriesClassifierConfiguration

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
@available
extension TimeSeriesClassifier Sendable where Scalar
Sendable Label Sendable

@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
@available
extension TimeSeriesClassifier Model Codable

    /// Encodes this value into the given encoder.
    ///
    /// If the value fails to encode anything, `encoder` will encode an empty
    /// keyed container in its place.
    ///
    /// This function throws an error if any values are invalid for the given
    /// encoder's format.
    ///
    /// - Parameter encoder: The encoder to write data to.
public func encode any Encoder throws

```

```
    /// Creates a new instance by decoding from the given decoder.  
    ///  
    /// This initializer throws an error if reading from the decoder fails, or  
    /// if the data read is corrupted or otherwise invalid.  
    ///  
    /// - Parameter decoder: The decoder to read data from.  
public init any Decoder throws  
  
extension TimeSeriesClassifier Model  
  
    /// Exports this transformer as a CoreML model package.  
    ///  
    /// - Parameter url: The location to write the model into.  
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
  
@available  
public func export URL throws  
  
    /// Exports this transformer as a CoreML model package with user-supplied  
metadata.  
    ///  
    /// - Parameters:  
    ///   - url: The location to write the model into.  
    ///   - metadata: Contextual user-provided information.  
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
  
@available  
public func export URL ModelMetadata  
throws  
  
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
@available  
extension TimeSeriesClassifier Model TemporalTransformer  
  
    /// The output async sequence type.  
public typealias OutputSequence  
AnyTemporalSequence ClassificationDistribution Label  
  
    /// Performs the transformation on an input sequence.  
    ///  
    /// - Parameters:  
    ///   - input: A temporal sequence of features. Each feature's shape  
must be `[featureSize]`.  
    ///   - eventHandler: An event handler.  
    /// - Returns: An temporal sequence of predictions. Each prediction's  
shape is  
    ///   ` [forecastWindowSize, annotationSize]`.
```

```
public func applied some
TemporalSequence MLShapedArray Scalar
EventHandler nil async throws
AnyTemporalSequence ClassificationDistribution Label

/// The configuration for a time-series classifier.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
@available
public struct TimeSeriesClassifierConfiguration Hashable
Codable Sendable

    /// The minimum number of samples required to produce a classification.
    ///
    /// This configuration parameter controls the size of the model. Set it to the
    largest value that is reasonable for
    /// your task. For example if your input is accelerometer data sampled at
    100Hz and you want to recognize actions,
    /// the minimum action duration may be one second, so the minimum
    sequence length is `100Hz * 1s = 100`. Defaults
    /// to 100.
    public var minimumSequenceLength Int

    /// The maximum number of samples that can be classified.
    ///
    /// This configuration parameter is used as the input size when exporting a
    Core ML model.
    public var maximumSequenceLength Int

    /// The number of examples in each training batch.
    ///
    /// - Note: This parameter is only used by the `fitted` method.
    public var batchSize Int

    /// The maximum number of allowed passes through the data.
    ///
    /// More passes over the data can result in a more accurately trained model.
    Consider increasing this if the
    /// training accuracy is low. Defaults to 25.
    ///
    /// - Note: This parameter is only used by the `fitted` method. When
    using the `update` method it's up to you
    /// to decide when to stop.
    public var maximumIterationCount Int

    /// The early-stopping tolerance.
    ///
    /// The tolerance is used by the `fitted` method to decide when progress
    is no longer being made, in which case the
    /// training process will stop before the specified maximum number of
```

```
iterations (known as early stopping).
```

```
    /// Significant progress happens when the validation loss decreases by at  
    /// least the tolerance.
```

```
    ///
```

```
    /// Defaults to 0.01.
```

```
    ///
```

```
    /// - Note: Early stopping only happens when using the `fitted`  
    /// method with validation data.
```

```
public var earlyStoppingTolerance Float
```

```
    /// The number of iterations to use when evaluating whether to stop early.
```

```
    ///
```

```
    /// The `fitted` method will stop if no significant progress is made for this  
    /// many iterations. Significant
```

```
    /// progress happens when the validation error decreases by at least  
    /// `convergenceThreshold`.
```

```
    ///
```

```
    /// - Note: Early stopping only happens when using the `fitted`  
    /// method with validation data.
```

```
public var earlyStoppingIterationCount Int
```

```
    /// The starting learning rate.
```

```
    ///
```

```
    /// Defaults to 0.005.
```

```
public var learningRate Float
```

```
    /// A seed to generate reproducible results from random operations.
```

```
public var randomSeed Int
```

```
    /// Creates a configuration.
```

```
public init
```

```
    /// Hashes the essential components of this value by feeding them into the  
    /// given hasher.
```

```
    ///
```

```
    /// Implement this method to conform to the `Hashable` protocol. The  
    /// components used for hashing must be the same as the components  
    /// compared
```

```
    /// in your type's `==` operator implementation. Call  
    /// `hasher.combine(_:)`
```

```
    /// with each of these components.
```

```
    ///
```

```
    /// - Important: In your implementation of `hash(into:)`,  
    /// don't call `finalize()` on the `hasher` instance provided,  
    /// or replace it with a different instance.
```

```
    /// Doing so may become a compile-time error in the future.
```

```
    ///
```

```
    /// - Parameter hasher: The hasher to use when combining the  
    /// components
```

```
    /// of this instance.
```

```
public func hash
```

```
inout Hasher
```

```
/// Returns a Boolean value indicating whether two values are equal.
```

```
///
```

```
/// Equality is the inverse of inequality. For any values `a` and `b`,  
/// `a == b` implies that `a != b` is `false`.
```

```
///
```

```
/// - Parameters:
```

```
///   - lhs: A value to compare.
```

```
///   - rhs: Another value to compare.
```

```
public static func
```

```
TimeSeriesClassifierConfiguration
```

```
TimeSeriesClassifierConfiguration Bool
```

```
/// Encodes this value into the given encoder.
```

```
///
```

```
/// If the value fails to encode anything, `encoder` will encode an empty  
/// keyed container in its place.
```

```
///
```

```
/// This function throws an error if any values are invalid for the given  
/// encoder's format.
```

```
///
```

```
/// - Parameter encoder: The encoder to write data to.
```

```
public func encode any Encoder throws
```

```
/// The hash value.
```

```
///
```

```
/// Hash values are not guaranteed to be equal across different executions of  
/// your program. Do not save hash values to use during a future execution.
```

```
///
```

```
/// - Important: `hashValue` is deprecated as a `Hashable`  
requirement. To
```

```
///   conform to `Hashable`, implement the `hash(into:)` requirement  
instead.
```

```
///   The compiler provides an implementation for `hashValue` for you.
```

```
public var hashValue Int get
```

```
/// Creates a new instance by decoding from the given decoder.
```

```
///
```

```
/// This initializer throws an error if reading from the decoder fails, or  
/// if the data read is corrupted or otherwise invalid.
```

```
///
```

```
/// - Parameter decoder: The decoder to read data from.
```

```
public init any Decoder throws
```

```
/// A sequence of forecasting windows on a time series shaped array.
```

```
///
```

```
/// A time-series forecaster takes a series of samples and produces a prediction of  
the next samples.
```

```

/// For example the sequence `[1, 2, 3, 4]` could predict `[5, 6]`.
///
/// The shape of each feature in the sequence is `[inputWindowSize,
/// featureSize]` and the shape of each annotation is
/// `[forecastWindowSize, annotationSize]`. The sequence will return
/// as many feature-annotation examples as fit in
/// the input. For example an input sequence of size of 10 with an input sample
/// count of 4, a prediction sample count
/// of 2, and a stride of 1 will produce 5 annotated windows:
///
/// ...
/// feature: [1, 2, 3, 4], annotation: [5, 6]
/// feature: [2, 3, 4, 5], annotation: [6, 7]
/// feature: [3, 4, 5, 6], annotation: [7, 8]
/// feature: [4, 5, 6, 7], annotation: [8, 9]
/// feature: [5, 6, 7, 8], annotation: [9, 10]
/// ...
///
/// Note that 9 and 10 are never used as features because there would be no
/// annotations for those samples.
@available(macOS 15.0, iOS 18.0, tvOS 18.0, visionOS 2.0,
watchOS 11.0)
public struct TimeSeriesForecasterAnnotatedWindows
Sequence @unchecked Sendable where Scalar : MLShapedArrayScalar

    /// A type representing the sequence's elements.
    public typealias Element = AnnotatedFeature<MLShapedArray<Scalar>, MLShapedArray<Scalar>>

    /// The original features.
    public let features: MLShapedArray<Scalar>

    /// The original annotations.
    public let annotations: MLShapedArray<Scalar>

    /// The input sample count.
    public let inputWindowSize: Int

    /// The prediction sample count.
    public let forecastWindowSize: Int

    /// The number of samples between windows.
    public var stride: Int

    /// A Boolean value indicating whether to shuffle the elements.
    public var shufflesElements: Bool

    /// Creates a batch sequence.
    ///

```

```

    /// - Parameters:
    ///   - features: A shaped array of features, it must have two
    dimensions.
    ///   - annotations: A shaped array of annotations, it must have two
    dimensions.
    ///   - inputWindowSize: The number of input samples. Must be
    positive.
    ///   - forecastWindowSize: The number of prediction samples. Must
    be positive.
    ///   - stride: The number of samples between windows. Must be
    positive. Defaults to 1.
    ///   - shufflesElements: A Boolean value indicating whether to
    shuffle the elements. Defaults to true.
public init MLShapedArray Scalar
MLShapedArray Scalar Int
Int Int 1
Bool true throws

    /// A value less than or equal to the number of elements in the sequence,
    /// calculated nondestructively.
    ///
    /// The default implementation returns 0. If you provide your own
    /// implementation, make sure to compute the value nondestructively.
    ///
    /// - Complexity: O(1), except if the sequence also conforms to
    `Collection`.
    /// In this case, see the documentation of
    `Collection.underestimatedCount`.
public var underestimatedCount Int get

    /// Returns an iterator over the elements of this sequence.
public func makeIterator
TimeSeriesForecasterAnnotatedWindows Scalar Iterator

extension TimeSeriesForecasterAnnotatedWindows

    /// A time-series forecaster batch sequence iterator.
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct Iterator : IteratorProtocol @unchecked
Sendable

    /// Advances to the next element and returns it, or `nil` if no next
element
    /// exists.
    ///
    /// Repeatedly calling this method returns, in order, all the elements of
the
    /// underlying sequence. As soon as the sequence has run out of
elements, all

```

been
to
and

```
    /// subsequent calls return `nil`.  
    ///  
    /// You must not call this method if any other copy of this iterator has  
    /// advanced with a call to its `next()` method.  
    ///  
    /// The following example shows how an iterator can be used explicitly  
    /// to  
    /// emulate a `for`-`in` loop. First, retrieve a sequence's iterator,  
    /// then call the iterator's `next()` method until it returns `nil`.  
    ///  
    ///     let numbers = [2, 3, 5, 7]  
    ///     var numbersIterator = numbers.makeIterator()  
    ///  
    ///         while let num = numbersIterator.next() {  
    ///             print(num)  
    ///         }  
    ///         // Prints "2"  
    ///         // Prints "3"  
    ///         // Prints "5"  
    ///         // Prints "7"  
    ///  
    /// - Returns: The next element in the underlying sequence, if a  
next element  
    /// exists; otherwise, `nil`.
```

public mutating func next

AnnotatedFeature MLShapedArray Scalar
MLShapedArray Scalar

```
    /// The type of element traversed by the iterator.  
    @available iOS 18.0 tvOS 18.0 watchOS 11.0 visionOS  
2.0 macOS 15.0  
    public typealias Element  
AnnotatedFeature MLShapedArray Scalar MLShapedArray Scalar
```

```
/// A sequence of forecaster batches on a time series shaped array.  
///  
/// A time-series forecaster takes a series of samples and produces a prediction of  
the next samples.  
/// For example the sequence `[1, 2, 3, 4]` could predict `[5, 6]`.  
/// To train a forecaster, each training batch contains the input samples along with  
the annotations (ground truth  
/// predictions). For example a batch could have this:  
///  
/// ` ` `  
/// features = [  
///     [1, 2, 3, 4],
```

```

///      [2, 3, 4, 5],
///      [3, 4, 5, 6],
/// ]
/// annotations = [
///      [5, 6],
///      [6, 7],
///      [7, 8],
/// ]
/// ````
/// 
/// The shape of the features in the sequence is `[batchSize,
/// inputWindowSize, featureSize]` and the shape of the
/// annotations is `[batchSize, forecastWindowSize,
/// annotationSize]`. The batch sequence will return as many
/// feature-annotation examples as fit in the input. For example, an input sequence
/// size of 10 with an input sample
/// count of 4 and a prediction sample count of 2 will produce 5 examples:
/// 
/// ````
/// features: [1, 2, 3, 4], annotations: [5, 6]
/// features: [2, 3, 4, 5], annotations: [6, 7]
/// features: [3, 4, 5, 6], annotations: [7, 8]
/// features: [4, 5, 6, 7], annotations: [8, 9]
/// features: [5, 6, 7, 8], annotations: [9, 10]
/// ````
/// 
/// Note that 9 and 10 are never used as features because there would be no
/// annotations for those examples.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
public struct TimeSeriesForecasterBatches : Sequence
@unchecked Sendable where Scalar : MLShapedArrayScalar

    /// A type representing the sequence's elements.
    public typealias Element = AnnotatedBatch<Scalar>

    /// The original features.
    public let features : MLShapedArray<Scalar>

    /// The original annotations.
    public let annotations : MLShapedArray<Scalar>

    /// The batch size.
    public let batchSize : Int

    /// The input sample count.
    public let inputWindowSize : Int

    /// The prediction sample count.
    public let forecastWindowSize : Int

```

```

    /// A Boolean value indicating whether to shuffle the batches.
public var shufflesBatches Bool

    /// Creates a batch sequence.
    ///
    /// - Parameters:
    ///   - features: A shaped array of features, it must have two dimensions.
    ///   - annotations: A shaped array of annotations, it must have two dimensions.
    ///   - batchSize: The batch size. Must be positive.
    ///   - inputWindowSize: The number of input samples. Must be positive.
    ///   - forecastWindowSize: The number of prediction samples. Must be positive.
    ///   - shufflesBatches: A Boolean value indicating whether to shuffle the batches.
public init MLShapedArray Scalar
MLShapedArray Scalar Int Int
Int Bool true throws

    /// A value less than or equal to the number of elements in the sequence,
    /// calculated nondestructively.
    ///
    /// The default implementation returns 0. If you provide your own
    /// implementation, make sure to compute the value nondestructively.
    ///
    /// - Complexity: O(1), except if the sequence also conforms to
    `Collection`.
    /// In this case, see the documentation of
    `Collection.underestimatedCount`.
public var underestimatedCount Int get

    /// Returns an iterator over the elements of this sequence.
public func makeIterator
TimeSeriesForecasterBatches Scalar Iterator

```

extension TimeSeriesForecasterBatches

```

    /// A time-series forecaster batch sequence iterator.
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
    public struct Iterator : IteratorProtocol @unchecked
Sendable

    /// Advances to the next element and returns it, or `nil` if no next
element
    /// exists.

```

```

    /**
     * Repeatedly calling this method returns, in order, all the elements of
the
     * underlying sequence. As soon as the sequence has run out of
elements, all
     * subsequent calls return `nil`.
    /**
     * You must not call this method if any other copy of this iterator has
been
     * advanced with a call to its `next()` method.
    /**
     * The following example shows how an iterator can be used explicitly
to
     * emulate a `for`-`in` loop. First, retrieve a sequence's iterator,
and
     * then call the iterator's `next()` method until it returns `nil`.
    /**
     * let numbers = [2, 3, 5, 7]
     * var numbersIterator = numbers.makeIterator()
    /**
     * while let num = numbersIterator.next() {
     *     print(num)
    }
    // Prints "2"
    // Prints "3"
    // Prints "5"
    // Prints "7"
    /**
     * - Returns: The next element in the underlying sequence, if a
next element
     * exists; otherwise, `nil`.
public mutating func next      AnnotatedBatch Scalar

     * The type of element traversed by the iterator.
@available iOS 18.0  tvOS 18.0  watchOS 11.0  visionOS
2.0  macOS 15.0
public typealias Element      AnnotatedBatch Scalar

```

```

     * A transformer that takes an input and produces an output.
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public protocol Transformer

```

```

     * The input type.
associatedtype Input

     * The output type.
associatedtype Output

```

```
    /// Performs the transformation on a single input.  
    ///  
    /// - Parameters:  
    ///   - input: The transformer input.  
    ///   - eventHandler: An event handler.  
    /// - Returns: An output produced by applying the transformer to the  
input.  
    func applied Self Input  
EventHandler async throws Self Output
```

extension Transformer

```
    /// Returns a random transformer wrapping a transformer.  
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
    public func adaptedAsRandomTransformer some  
RandomTransformer Self Input Self Output
```

extension Transformer

```
    /// Composes this transformer with a temporal transformer.  
    @available 13.0 15.0  
    @available 16.0 18.0  
    @available 16.0 18.0  
    @available 1.0 2.0  
    @available  
    public func appending Other _ Other  
ComposedTemporalTransformer TransformerToTemporalAdaptor Self  
Other where Other TemporalTransformer Self Output  
Other Input  
  
    /// Composes this transformer with a temporal transformer.  
    @available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0  
watchOS 11.0  
    public func appending Other _ Other  
ComposedTemporalTransformer TemporalAdaptor Self Other  
where Self Sendable Other TemporalTransformer  
Self Output Other Input
```

extension Transformer

```
    /// Composes this transformer with another transformer.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
  
    public func appending Other _ Other
```

```
ComposedTransformer Self Other where Other Transformer
Self Output Other Input

    /// Composes this transformer with an annotated-feature transformer.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func appending Other Annotation _ Other
        some Transformer AnnotatedFeature Self Input Annotation
    Other Output where Other Transformer Other Input
    AnnotatedFeature Self Output Annotation

    /// Composes this transformer with a feature-only transformer.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func appending Other Annotation _ Other
        some Transformer Self Input AnnotatedFeature Other Output
    Annotation where Other Transformer Self Output
    AnnotatedFeature Other Input Annotation

    /// Composes this transformer with an annotated-feature transformer.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func appending Other Annotation _ Other
        some Transformer AnnotatedPrediction Self Input
    Annotation Other Output where Other Transformer
    Other Input AnnotatedPrediction Self Output Annotation

    /// Composes this transformer with a prediction-only transformer.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

        public func appending Other Annotation _ Other
        some Transformer Self Input
    AnnotatedPrediction Other Output Annotation where Other
    Transformer Self Output AnnotatedPrediction Other Input
    Annotation

extension Transformer

    /// Composes this transformer with an estimator.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

        public func appending Other _ Other
    PreprocessingEstimator Self Other where Other Estimator
    Self Output Other Transformer Input
```

extension Transformer

```
    /// Composes this transformer with an updatable supervised temporal
estimator.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    @available(13.0, 15.0)
    @available(16.0, 18.0)
    @available(16.0, 18.0)
    @available(1.0, 2.0)
    @available
    public func appending<Other>_<Other>
    PreprocessingUpdatableSupervisedTemporalEstimator<TransformerT>
    oTemporalAdaptor<Self, Other> where Other : UpdatableSupervisedTemporalEstimator<Self, Output>
    Output : Transformer<Input>
```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0 **extension** Transformer

```
    /// Performs the transformation on a sequence of inputs.
    ///
    /// - Parameters:
    ///   - input: The transformer inputs.
    ///   - eventHandler: An event handler.
    /// - Returns: The outputs produced by applying the transformer to the
inputs.
    @inlinable public func applied<S>_<S>
        EventHandler<nil, async throws>
    Self<Output> where S : Sequence<Self<Input>>, S.Element ==
```



```
    /// Performs the transformation on a sequence of annotated inputs.
    ///
    /// - Parameters:
    ///   - input: A sequence of annotated inputs.
    ///   - eventHandler: An event handler.
    /// - Returns: The annotated outputs produced by applying the
transformer to the inputs.
    @inlinable public func applied<S, Annotation>_<S>
        EventHandler<nil, async throws>
    AnnotatedFeature<Self<Output>, Annotation> where S : Sequence<S.Element>, S.Element == AnnotatedFeature<Self<Input>, Annotation>
```



```
    /// Performs a prediction on a sequence of annotated inputs.
    ///
    /// - Parameters:
    ///   - input: A sequence of annotated inputs.
    ///   - eventHandler: An event handler.
```

```

    /// - Returns: Annotated predictions produced by applying the
transformer to the inputs.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

@inlinable public func prediction S Annotation
    S EventHandler nil async throws
AnnotatedPrediction Self Output Annotation where S
Sequence S Element AnnotatedFeature Self Input
Annotation

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension Transformer

    /// Performs the transformation on a single input.
    ///
    /// - Parameters:
    ///   - input: The transformer input.
    ///   - eventHandler: An event handler.
    /// - Returns: An output produced by applying the transformer to the
input.
    @inlinable public func callAsFunction _ Self Input
        EventHandler nil async throws Self Output

    /// Performs the transformation on a sequence of inputs.
    ///
    /// - Parameters:
    ///   - input: The transformer inputs.
    ///   - eventHandler: An event handler.
    /// - Returns: The outputs produced by applying the transformer to the
inputs.
    @inlinable public func callAsFunction S _
        EventHandler nil async throws
Self Output where S Sequence Self Input S Element

extension Transformer

    /// Exports this transformer as a CoreML model.
    ///
    /// - Note: By default this method exports .mlpackage files. You can
export a .mlmodel file by specifying that as
    /// the URL file extension. But if you specify .mlmodel and the transformer
doesn't support it, this method will
    /// throw an error.
    ///
    /// - Parameter url: The location to write the model into.
@available macOS 13.0 iOS 16.0 tvOS 16.0
public func export URL throws

```

```

    /// Exports this transformer as a CoreML model with userInfo.
    ///
    /// - Note: By default this method exports .mlpackage files. You can
    export a .mlmodel file by specifying that as
    /// the URL file extension. But if you specify .mlmodel and the transformer
    doesn't support it, this method will
    /// throw an error.
    ///
    /// - Parameters:
    ///   - url: The location to write the model into.
    ///   - metadata: Contextual user-provided information.
    @available macOS 14.0 iOS 17.0 tvOS 17.0
    public func export(URL ModelMetadata)
throws

```

extension Transformer

```

    /// Composes this transformer with an updatable temporal estimator.
    @available 13.0 15.0
    @available 16.0 18.0
    @available 16.0 18.0
    @available 1.0 2.0
    @available
    public func appending(Other) -> Other
    PreprocessingUpdatableTemporalEstimator TransformerToTemporalA
    daptor Self Other where Other : UpdatableTemporalEstimator
    Self Output Other Transformer Input

```

extension Transformer

```

    /// Returns an annotated-prediction transformer that transforms the
    predictions using this transformer while
    /// leaving the annotations unchanged.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

    public func
    adaptedAsAnnotatedPredictionTransformer(Annotation)
        Annotation self some
    Transformer AnnotatedPrediction Self Input Annotation
    AnnotatedPrediction Self Output Annotation

```

extension Transformer

```

    /// Composes this transformer with a supervised temporal estimator.
    @available 13.0 15.0
    @available 16.0 18.0

```

```
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other _ Other
PreprocessingSupervisedTemporalEstimator TransformerToTemporal
Adaptor Self Other where Other
SupervisedTemporalEstimator Self Output
Other Transformer Input
```

extension Transformer

```
/// Exposes this transformer as a temporal transformer.
@available macOS 15.0 iOS 18.0 tvOS 18.0 visionOS 2.0
watchOS 11.0
@inlinable @preconcurrency public func adaptedAsTemporal
TemporalAdaptor Self where Self Sendable
```

extension Transformer

```
/// Returns an annotated-feature transformer that transforms the features
using this transformer while leaving the
/// annotations unchanged.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func
adaptedAsAnnotatedFeatureTransformer Annotation
Annotation self some
Transformer AnnotatedFeature Self Input Annotation
AnnotatedFeature Self Output Annotation
```

extension Transformer

```
/// Composes this transformer with a supervised estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func appending Other _ Other
PreprocessingSupervisedEstimator Self Other where Other
SupervisedEstimator Self Output Other Transformer Input
```

extension Transformer

```
/// Exposes this transformer as a temporal transformer.
@available 13.0 15.0
@available 16.0 18.0
```

<code>@available</code>	16.0	18.0
<code>@available</code>	1.0	2.0
<code>@available</code>		
<code>@inlinable public func adaptedAsTemporal</code>		
<code>TransformerToTemporalAdaptor Self</code>		

extension Transformer

```
/// Composes this transformer with an updatable estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func appending Other - Other
PreprocessingUpdatableEstimator Self Other where Other UpdatableEstimator Self Output Other Transformer Input
```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

extension Transformer

```
/// Performs a prediction from a single input.
///
/// - Parameter input: The input feature.
/// - Returns: A classification array.
public func prediction Label -> Self Input
async throws ClassificationDistribution Label where Label Hashable Self Output ClassificationDistribution Label
```

```
/// Performs a prediction from a sequence of inputs.
///
/// - Parameter input: The input features.
/// - Returns: An array of classification distributions.
public func prediction S Label -> S Input
throws ClassificationDistribution Label where S Sequence Label Hashable Self Input S Element Self Output ClassificationDistribution Label
```

extension Transformer

```
/// Exposes this transformer as a trivial estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func adaptedAsEstimator
TransformerToEstimatorAdaptor Self
```

extension Transformer

```

    /// Exposes this transformer as a trivial estimator.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)

    @inlinable public func adaptedAsUpdatableEstimator
        TransformerToUpdatableEstimatorAdaptor<Self>

extension Transformer
    /// Composes this transformer with an updatable supervised estimator.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)

    public func appending<Other: UpdatableSupervisedEstimator>(_ other: Other)
        PreprocessingUpdatableSupervisedEstimator<Self, Other> where
            Other: UpdatableSupervisedEstimator, Self: Output<Other>, Other: Transformer<Input>

extension Transformer
    /// Composes this transformer with a temporal estimator.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 1.0, 2.0)
    @available(macOS 15.0, iOS 18.0, tvOS 18.0)

    public func appending<Other: TemporalEstimator>(_ other: Other)
        PreprocessingTemporalEstimator<TransformerToTemporalAdaptor<Self, Other>> where
            Other: TemporalEstimator, Self: Output<Other>, Other: Transformer<Input>

    /// An estimator that always returns a predefined transformer.
    @available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
    public struct TransformerToEstimatorAdaptor<T: Transformer>: Estimator where T: Transformer

    /// A pre-defined transformer.
    public let transformer: T

    /// Creates a trivial estimator.
    public init<T: Transformer>(_ transformer: T)
        /// Returns the pre-defined transformer.
        ///
        /// - Parameters:
        ///   - input: A sequence of examples.
        ///   - eventHandler: An event handler.

```

```

    /// - Returns: The pre-defined transformer.
    @inlinable public func fitted S           S
        EventHandler nil async throws Transformer
    where S Sequence Transformer Input S Element

    /// Does nothing since this estimator uses a pre-defined transformer.
    @inlinable public func encode _           Transformer
        inout any EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    @inlinable public func decode             inout any
EstimatorDecoder throws Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TransformerToEstimatorAdaptor Sendable where
Transformer Sendable

/// A temporal transformer that applies a regular transformer to each value of a
temporal sequence.
@available 13.0 15.0
"TemporalAdaptor"
@available 16.0 18.0
"TemporalAdaptor"
@available 16.0 18.0
"TemporalAdaptor"
@available 1.0 2.0
"TemporalAdaptor"
@available
public struct TransformerToTemporalAdaptor Base
TemporalTransformer where Base Transformer

    /// The input type.
    public typealias Input Base Input

    /// The output type.
    public typealias Output Base Output

    /// The output sequence type.
    public typealias OutputSequence
AnyTemporalSequence TransformerToTemporalAdaptor Base Output

    /// Creates a temporal transformer from a transformer.
    ///
    /// The resulting transformer applies the underlying transformer to each
element in the input sequence.
    public init _           Base

    /// Performs the transformation on each element of the input sequence.

```

```

@inlinable public func applied S           S
    EventHandler nil async throws
AnyTemporalSequence TransformerToTemporalAdaptor Base Output
where S TemporalSequence Base Input S Feature

@available           13.0      15.0
@available           16.0      18.0
@available           16.0      18.0
@available           1.0       2.0
@available
extension TransformerToTemporalAdaptor Sendable where Base
Sendable

/// An updatable estimator that always returns a predefined transformer.
@available macOS 13.0  iOS 16.0  tvOS 16.0  watchOS 11.0
public struct
TransformerToUpdatableEstimatorAdaptor Transformer
UpdatableEstimator where Transformer Transformer

/// A pre-defined transformer.
public let transformer Transformer

/// Creates a trivial estimator.
public init _ Transformer

/// Creates a default-initialized transformer suitable for incremental fitting.
@inlinable public func makeTransformer Transformer

/// Returns the pre-defined transformer.
///
/// - Parameters:
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
/// - Returns: The pre-defined transformer.
@inlinable public func fitted S           S
    EventHandler nil async throws Transformer
where S Sequence Transformer Input S Element

/// Does nothing since this estimator uses a pre-defined transformer.
@inlinable public func update InputSequence -
    inout Transformer InputSequence
    EventHandler nil async throws where
InputSequence Sequence Transformer Input
InputSequence Element

/// Does nothing since this estimator uses a pre-defined transformer.
@inlinable public func encode _ Transformer

```

```

        inout any EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    @inlinable public func decode             inout any
EstimatorDecoder throws      Transformer

    /// Does nothing since this estimator uses a pre-defined transformer.
    @inlinable public func encodeWithOptimizer _ 
Transformer           inout any EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    @inlinable public func decodeWithOptimizer
inout any EstimatorDecoder throws      Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TransformerToUpdatableEstimatorAdaptor Sendable
where Transformer Sendable

/// A trained tree classifier model.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct TreeClassifierModel Label TabularTransformer
Sendable where Label Hashable

/// The names of the columns containing feature values.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var featureColumnNames String get

/// The name of the column containing the predicted labels.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var predictionColumnName String

/// The number of classes.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public var classCount Int get

/// Performs a classification on a data frame.
///
/// - Parameters:
///   - input: The input data frame.
///   - eventHandler: An event handler.
/// - Returns: A data frame of classifications.
public func applied DataFrame
EventHandler nil async throws DataFrame

```

```

    /// Builds a data frame containing a labels column and a probability
    distribution column.
    public func buildDataFrame _  

ClassificationDistribution Label DataFrame

    /// The input type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Input DataFrame

    /// The output type.
    @available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0

    public typealias Output DataFrame

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension TreeClassifierModel CustomDebugStringConvertible

    /// A textual representation of this instance, suitable for debugging.
    ///
    /// Calling this property directly is discouraged. Instead, convert an
    /// instance of any type to a string by using the `String(reflecting:)`  

    /// initializer. This initializer works with any type, and uses the custom
    /// `debugDescription` property for types that conform to
    /// `CustomDebugStringConvertible`:
    ///
    ///     struct Point: CustomDebugStringConvertible {
    ///         let x: Int, y: Int
    ///         var debugDescription: String {
    ///             return "(\(x), \(y))"
    ///         }
    ///     }
    ///
    ///     let p = Point(x: 21, y: 30)
    ///     let s = String(reflecting: p)
    ///     print(s)
    ///     // Prints "(21, 30)"
    ///
    /// The conversion of `p` to a string in the assignment to `s` uses the
    /// `Point` type's `debugDescription` property.
    public var debugDescription String get

    /// A trained tree regressor model.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct TreeRegressorModel TabularTransformer  

Sendable

```

```
/// The names of the columns containing feature values.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public var featureColumnNames String get  
  
/// The name of the column containing the predicted values.  
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0  
  
public var predictionColumnName String  
  
/// Performs a regression on a data frame.  
///  
/// - Parameters:  
///   - input: The regressor input.  
///   - eventHandler: An event handler.  
/// - Returns: A data frame of regressions.  
public func applied DataFrame  
EventHandler nil async throws DataFrame  
  
/// The input type.  
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
public typealias Input DataFrame  
  
/// The output type.  
@available iOS 16.0 tvOS 16.0 watchOS 11.0 macOS 13.0  
  
public typealias Output DataFrame  
  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
extension TreeRegressorModel : CustomDebugStringConvertible  
  
/// A textual representation of this instance, suitable for debugging.  
///  
/// Calling this property directly is discouraged. Instead, convert an  
/// instance of any type to a string by using the `String(reflecting:)`  
/// initializer. This initializer works with any type, and uses the custom  
/// `debugDescription` property for types that conform to  
/// `CustomDebugStringConvertible`:  
///  
///     struct Point: CustomDebugStringConvertible {  
///         let x: Int, y: Int  
///  
///         var debugDescription: String {  
///             return "(\(x), \(y))"  
///         }  
///     }  
/// }
```

```


    /**
     *      let p = Point(x: 21, y: 30)
     *      let s = String(reflecting: p)
     *      print(s)
     *      // Prints "(21, 30)"
     *
     *      /// The conversion of `p` to a string in the assignment to `s` uses the
     *      /// `Point` type's `debugDescription` property.
     public var debugDescription String get

    /// Applies the transformer with a randomly generated input parameter.
    ///
    /// The parameter is chosen from a continuous uniform distribution in the specified
    range.
    ///
    /// Note that a new transformer is created every time this transformer is applied.
    /// This may cause performance issues if the embedded transformer creation is
    costly.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct
UniformRandomFloatingPointParameter RandomTransformer
Parameter RandomTransformer where RandomTransformer
RandomTransformer Parameter BinaryFloatingPoint
RandomTransformer Input RandomTransformer Output
Parameter RawSignificand FixedWidthInteger

    /// The range of a random number to use as input to the transformer.
    public var range ClosedRange Parameter

    /// Creates a Random Parameter transformer.
    ///
    /// - Parameters:
    ///   - range: The range of a random number to use as input to the
    transformer.
    ///   - augmentation: An augmentation builder.
    public init Input ClosedRange Parameter
    AugmentationBuilder Input _ @escaping
    Parameter RandomTransformer where Input
    RandomTransformer Input

    /// Performs the random apply operation on the input.
    /// - Parameters:
    ///   - input: An input.
    ///   - generator: A random number generator.
    ///   - eventHandler: An event handler.
    /// - Returns: The randomly transformed image.
    public func applied RandomTransformer Input
        inout some RandomNumberGenerator
    EventHandler nil async throws RandomTransformer Output


```

```

    /// The input type.
    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Input RandomTransformer Input

    /// The output type.
    @available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0

    public typealias Output RandomTransformer Input

    /// Applies the transformer with a randomly generated input parameter.
    ///
    /// The parameter is chosen from a discrete uniform distribution in the specified
    range.
    ///
    /// Note that a new transformer is created every time this transformer is applied.
    /// This may cause performance issues if the embedded transformer creation is
    costly.
    @available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
    final public class
        UniformRandomIntegerParameter RandomTransformer Parameter
        RandomTransformer where RandomTransformer RandomTransformer
        Parameter FixedWidthInteger RandomTransformer Input
        RandomTransformer Output

    /// The range of a random number to use as input to the transformer.
    final public var range Range Parameter

    /// Creates a Random Parameter transformer.
    ///
    /// - Parameters:
    ///   - range: The range of a random number to use as input to the
    transformer.
    ///   - augmentation: An augmentation builder.
    public init Input Range Parameter
        AugmentationBuilder Input _ @escaping
        Parameter RandomTransformer where Input
        RandomTransformer Input

    /// Creates a Random Parameter transformer.
    ///
    /// - Parameters:
    ///   - range: The range of a random number to use as input to the
    transformer.
    ///   - augmentation: An augmentation builder.
    public init Input ClosedRange Parameter
        AugmentationBuilder Input _ @escaping
        Parameter RandomTransformer where Input

```

RandomTransformer Input

```
/// Performs the random apply operation on the input.  
/// - Parameters:  
///   - input: An input.  
///   - generator: A random number generator.  
///   - eventHandler: An event handler.  
/// - Returns: The randomly transformed image.  
final public func applied  
RandomTransformer Input           inout some  
RandomNumberGenerator           EventHandler      nil  
async throws     RandomTransformer Output  
  
/// The input type.  
@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0  
  
public typealias Input = RandomTransformer Input  
  
/// The output type.  
@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS 14.0  
  
public typealias Output = RandomTransformer Input  
  
/// An estimator that can be incrementally updated.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0  
public protocol UpdatableEstimator : Estimator  
  
/// Creates a default-initialized transformer suitable for incremental fitting.  
func makeTransformer : Self Transformer  
  
/// Updates a transformer with a new sequence of examples.  
///  
/// - Parameters:  
///   - transformer: A transformer to update.  
///   - input: A sequence of examples.  
///   - eventHandler: An event handler.  
func update InputSequence _           inout  
Self Transformer           InputSequence  
EventHandler    async throws where InputSequence : Sequence  
InputSequence Element     Self Transformer Input  
  
/// Encodes the transformer and optimizer to an encoder.  
///  
/// - Parameters:  
///   - transformer: A transformer this estimator creates.  
///   - encoder: An encoder.  
func encodeWithOptimizer _           Self Transformer  
          inout any EstimatorEncoder throws
```

```

/// Reads the encoded transformer and optimizer with a decoder.
///
/// - Parameter decoder: A decoder.
/// - Returns: The decoded transformer.
func decodeWithOptimizer inout any
EstimatorDecoder throws Self Transformer

extension UpdatableEstimator

/// Composes this updatable estimator with a temporal transformer.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other - Other some
UpdatableTemporalEstimator ComposedTemporalTransformer Transfo
rmerToTemporalAdaptor Self Transformer Other where Other
TemporalTransformer Other Input Self Transformer Output

/// Composes this updatable estimator with an updatable temporal estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public func appending Other - Other some
UpdatableTemporalEstimator ComposedTemporalTransformer Transfo
rmerToTemporalAdaptor Self Transformer Other Transformer
where Other UpdatableTemporalEstimator
Self Transformer Output Other Transformer Input

extension UpdatableEstimator

/// Exposes this estimator as a supervised estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func
adaptedAsSupervised Annotation
Annotation self
UpdatableEstimatorToSupervisedAdaptor Self Annotation where
Annotation Equatable

```

```
extension UpdatableEstimator
```

```
    /// Composes this updatable estimator with a transformer.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        public func appending Other _ Other some
```

```
UpdatableEstimator ComposedTransformer Self Transformer
```

```
Other where Other Transformer Other Input
```

```
Self Transformer Output
```

```
    /// Composes this updatable estimator with another updatable estimator.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        public func appending Other _ Other some
```

```
UpdatableEstimator ComposedTransformer Self Transformer
```

```
Other Transformer where Other UpdatableEstimator
```

```
Self Transformer Output Other Transformer Input
```

```
extension UpdatableEstimator
```

```
    /// Composes this updatable estimator with an updatable supervised estimator.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        public func appending Other _ Other some
```

```
UpdatableSupervisedEstimator ComposedTransformer Self Transformer
```

```
Other Transformer Other Annotation where Other UpdatableSupervisedEstimator
```

```
Self Transformer Output Other Transformer Input
```

```
extension UpdatableEstimator
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        @inlinable public func update InputSequence _  
            inout Self Transformer
```

```
InputSequence async throws where InputSequence Sequence
```

```
InputSequence Element Self Transformer Input
```

```
extension UpdatableEstimator
```

```
    /// Composes this updatable estimator with an updatable supervised temporal estimator.
```

```
    @available
```

```
        13.0
```

```
        15.0
```

```

@available(16.0, 18.0)
@available(16.0, 18.0)
@available(1.0, 2.0)
@available(some, 2.0)
public func appending<Other>(_ other: Other) -> ComposedTemporalTransformer<Other, Self, TransformerToTemporalAdaptor<Other, OtherAnnotation, UpdatableSupervisedTemporalEstimator<Other, Self, TransformerOutput, OtherTransformerInput>>

```

extension UpdatableEstimator

```

/// Exposes this estimator as a temporal estimator.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0, 15.0)
@available(macOS 16.0, iOS 18.0, tvOS 18.0, watchOS 18.0, 18.0)
@available(macOS 16.0, iOS 1.0, tvOS 2.0, watchOS 2.0, 2.0)
@available(inlinable, 15.0)
public func adaptedAsTemporal<Self: UpdatableEstimatorToTemporalAdaptor<Self, EstimatorAnnotation, UpdatableSupervisedEstimator<EstimatorAnnotation, Equatable>>()

```

/// An adaptor that exposes an updatable estimator as an updatable supervised estimator.

```

@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0, 15.0)
public struct UpdatableEstimatorToSupervisedAdaptor<EstimatorAnnotation, UpdatableSupervisedEstimator<EstimatorAnnotation, Equatable>>: UpdatableEstimator<EstimatorAnnotation>

```

/// The transformer type created by this estimator.

```

public typealias Transformer = EstimatorTransformer<Estimator>

```

/// The wrapped estimator.

```

public let estimator: Estimator

```

/// Creates an estimator adaptor.

```

public init<_Estimator: Estimator>()

```

/// Creates a default-initialized transformer suitable for incremental fitting.

```

public func makeTransformer() -> EstimatorTransformer<Estimator>

```

/// Fits a transformer to a sequence of examples, ignoring the annotations and the validation.

```

/// - Parameters:
///   - input: A sequence of examples.
///   - eventHandler: An event handler.

```

```

    /// - Returns: The pre-defined transformer.
    public func fitted Input Input
EventHandler nil async throws
UpdatableEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where Input Sequence Input Element
AnnotatedFeature Estimator Transformer Input Annotation

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
    fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted Input Validation Input
Validation EventHandler
nil async throws
UpdatableEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where Input Sequence Validation
Sequence Input Element
AnnotatedFeature Estimator Transformer Input Annotation
Validation Element
AnnotatedFeature Estimator Transformer Input Annotation

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    public func update InputSequence _ inout
Estimator Transformer InputSequence
EventHandler nil async throws where
InputSequence Sequence InputSequence Element
AnnotatedFeature Estimator Transformer Input Annotation

    /// Fits a transformer to a sequence of examples while validating with a
    validation sequence.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - validation: A sequence of examples used for validation.
    ///   - eventHandler: An event handler.
    public func update InputSequence Validation _ inout
Estimator Transformer
InputSequence Validation
EventHandler nil async throws where
InputSequence Sequence Validation Sequence

```

```

InputSequence Element
AnnotatedFeature Estimator Transformer Input Annotation
Validation Element
AnnotatedFeature Estimator Transformer Input Annotation

    /// Does nothing since this estimator uses a pre-defined transformer.
    public func encode _  

UpdatableEstimatorToSupervisedAdaptor Estimator  

Annotation Transformer inout any  

EstimatorEncoder throws

    /// Returns the pre-defined transformer.
    public func decode inout any  

EstimatorDecoder throws  

UpdatableEstimatorToSupervisedAdaptor Estimator  

Annotation Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ////  

    /// - Parameters:  

    ///   - transformer: A transformer created by this estimator.  

    ///   - encoder: An encoder.  

    public func encodeWithOptimizer _  

UpdatableEstimatorToSupervisedAdaptor Estimator  

Annotation Transformer inout any  

EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer.
    ////  

    /// - Parameter decoder: A decoder.  

    /// - Returns: The decoded transformer.  

    public func decodeWithOptimizer inout any  

EstimatorDecoder throws  

UpdatableEstimatorToSupervisedAdaptor Estimator  

Annotation Transformer

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension UpdatableEstimatorToSupervisedAdaptor Sendable
where Estimator Sendable Annotation Sendable

    /// An updatable temporal estimator wrapping an updatable estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public struct UpdatableEstimatorToTemporalAdaptor Base

```

```
UpdatableTemporalEstimator where Base UpdatableEstimator

    /// The transformer type created by this estimator.
    public typealias Transformer
    UpdatableEstimatorToTemporalAdaptor Base Transformer

    /// The input type.
    public typealias Input Base Transformer Input

    /// The output type.
    public typealias Output Base Transformer Output

    /// Creates a temporal estimator from an estimator.
    ///
    /// The resulting estimator collects all elements of the input sequence before
    calling fit on the underlying
    /// estimator. The transformer returned from fit is also converted to a
    temporal transformer.
    public init _ Base

    /// Creates a default-initialized transformer suitable for incremental fitting.
    @inlinable public func makeTransformer
    UpdatableEstimatorToTemporalAdaptor Base Transformer

    /// Fits a transformer to a sequence of examples.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    @inlinable public func fitted InputSequence
    InputSequence EventHandler nil async throws
        UpdatableEstimatorToTemporalAdaptor Base Transformer where
        InputSequence Sequence InputSequence Element
        TemporalSequence Base Transformer Input
        InputSequence Element Feature

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    @inlinable public func update InputSequence _ inout
    UpdatableEstimatorToTemporalAdaptor Base Transformer
        InputSequence EventHandler nil async throws
        where InputSequence Sequence InputSequence Element
        TemporalSequence Base Transformer Input
        InputSequence Element Feature
```

```

    /// Encodes a fitted transformer.
    @inlinable public func encode _  

UpdatableEstimatorToTemporalAdaptor Base Transformer  

        inout any EstimatorEncoder throws

    /// Decodes the transformer.
    @inlinable public func decode           inout any  

EstimatorDecoder throws
UpdatableEstimatorToTemporalAdaptor Base Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ////  

    /// - Parameters:  

    ///   - transformer: A transformer created by this estimator.  

    ///   - encoder: An encoder.
    @inlinable public func encodeWithOptimizer _  

UpdatableEstimatorToTemporalAdaptor Base Transformer  

        inout any EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ////  

    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    @inlinable public func decodeWithOptimizer  

inout any EstimatorDecoder throws
UpdatableEstimatorToTemporalAdaptor Base Transformer

```

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		
extension UpdatableEstimatorToTemporalAdaptor	Sendable	where
Base	Sendable	

```

    /// A supervised estimator that can be incrementally updated.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol UpdatableSupervisedEstimator
    SupervisedEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
    func makeTransformer      Self Transformer

    /// Updates a transformer with a new sequence of examples.
    ////  

    /// - Parameters:

```

```

    /// - transformer: A transformer to update.
    /// - input: A sequence of examples.
    /// - eventHandler: An event handler.
func update InputSequence inout
Self Transformer InputSequence
EventHandler async throws where InputSequence Sequence
InputSequence Element
AnnotatedFeature Self Transformer Input Self Annotation

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
func encodeWithOptimizer inout any EstimatorEncoder throws
Self Transformer

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
func decodeWithOptimizer inout any
EstimatorDecoder throws Self Transformer

extension UpdatableSupervisedEstimator

    /// Composes this updatable estimator with a transformer.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other inout Other some
UpdatableSupervisedEstimator ComposedTransformer Self Transformer
Other Self Annotation where Other Transformer
Other Input Self Transformer Output

    /// Composes this updatable estimator with an updatable estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other inout Other some
UpdatableSupervisedEstimator ComposedTransformer Self Transformer
Other Transformer Self Annotation where Other
UpdatableEstimator Self Transformer Output
Other Transformer Input

    /// Composes this updatable estimator with an updatable supervised
estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

```

```
public func appending Other _          Other      some
UpdatableSupervisedEstimator ComposedTransformer Self Transformer
Other Transformer Self Annotation where Other
UpdatableSupervisedEstimator Self Annotation
Other Annotation Self Transformer Output
Other Transformer Input
```

extension UpdatableSupervisedEstimator

```
/// Composes this updatable supervised estimator with a temporal
transformer.
```

```
@available           13.0        15.0
@available           16.0        18.0
@available           16.0        18.0
@available           1.0         2.0
@available
@preconcurrency public func appending Other _ -
Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer
TransformerToTemporalAdaptor Self Transformer Other
Self Annotation where Other TemporalTransformer
Self Annotation Sendable Other Input
Self Transformer Output
```

```
/// Composes this updatable supervised estimator with an updatable
temporal estimator.
```

```
@available           13.0        15.0
@available           16.0        18.0
@available           16.0        18.0
@available           1.0         2.0
@available
@preconcurrency public func appending Other _ -
Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer
TransformerToTemporalAdaptor Self Transformer
Other Transformer Self Annotation where Other
UpdatableTemporalEstimator Self Annotation Sendable
Self Transformer Output Other Transformer Input
```

```
/// Composes this updatable supervised estimator with an updatable
supervised temporal estimator.
```

```
@available           13.0        15.0
@available           16.0        18.0
@available           16.0        18.0
@available           1.0         2.0
```

```
@available
public func appending Other _ Other some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer TransformerToTemporalAdaptor Self Transformer
Other Transformer Self Annotation where Other
UpdatableSupervisedTemporalEstimator Self Annotation
Other Annotation Self Transformer Output
Other Transformer Input
```

```
extension UpdatableSupervisedEstimator where Self Annotation Sendable
```

```
/// Exposes this supervised estimator as a temporal supervised estimator.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
@inlinable public func adaptedAsTemporal
UpdatableSupervisedEstimatorToTemporalAdaptor Self
```

```
available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension UpdatableSupervisedEstimator
```

```
/// Writes the encoded transformer and optimizer to a file.
///
/// - Parameters:
///   - transformer: A transformer created by this estimator.
///   - url: A file URL.
///   - overwrite: A Boolean value indicating whether to overwrite existing files.
public func writeWithOptimizer -
Self Transformer URL Bool true throws

/// Reads the encoded transformer and optimizer from a file.
///
/// - Parameter url: A file URL.
/// - Returns: The decoded transformer.
public func readWithOptimizer URL throws
Self Transformer
```

```
extension UpdatableSupervisedEstimator
```

```
available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```

@inlinable public func update InputSequence _  

    inout Self Transformer  

InputSequence async throws where InputSequence Sequence  

InputSequence Element  

AnnotatedFeature Self Transformer Input Self Annotation

extension UpdatableSupervisedEstimator

    /// Updates a transformer on an async sequence of examples.  

    ///  

    /// Note that the async sequence is collected before updating the  

transformer.  

    ///  

    /// - Parameters:  

    /// - transformer: A transformer to update.  

    /// - input: An async sequence of examples used for updating the  

transformer.  

    /// - eventHandler: An event handler.  

@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0

public func update Input inout  

Self Transformer Input  

EventHandler nil async throws where Input AsyncSequence  

Input Element AnnotatedFeature Self Transformer Input  

Self Annotation

    /// An updatable supervised temporal estimator wrapping an updatable supervised  

estimator.  

@available 13.0 15.0  

@available 16.0 18.0  

@available 16.0 18.0  

@available 1.0 2.0  

@available  

public struct  

UpdatableSupervisedEstimatorToTemporalAdaptor Base  

UpdatableSupervisedTemporalEstimator where Base  

UpdatableSupervisedEstimator Base Annotation Sendable

    /// The transformer type created by this estimator.  

public typealias Transformer  

TransformerToTemporalAdaptor Base Transformer

    /// The input type.  

public typealias Input Base Transformer Input

    /// The output type.  

public typealias Output Base Transformer Output

```

```

    ///> The annotation type.
public typealias Annotation Base Annotation

    ///> Creates a temporal supervised estimator from a supervised estimator.
    ///
    ///> The resulting estimator collects all elements of the input sequence before
    calling fit on the underlying
    ///> estimator. The transformer returned from fit is also converted to a
    temporal transformer.
public init _ Base

    ///> Creates a default-initialized transformer suitable for incremental fitting.
@inlinable public func makeTransformer
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer

    ///> Fits a transformer to a sequence of examples.
    ///
    ///> - Parameters:
    ///>   - input: A sequence of examples used for fitting the transformer.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
@inlinable public func fitted InputSequence
FeatureSequence InputSequence
EventHandler nil async throws
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer
where InputSequence Sequence FeatureSequence
TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Base Annotation
FeatureSequence Feature Base Transformer Input

    ///> Fits a transformer to a sequence of examples while validating with a
    validation sequence.
    ///
    ///> - Parameters:
    ///>   - input: A sequence of examples used for fitting the transformer.
    ///>   - validation: A sequence of examples used for validating the
    fitted transformer.
    ///>   - eventHandler: An event handler.
    ///> - Returns: The fitted transformer.
@inlinable public func fitted InputSequence Validation
FeatureSequence InputSequence
Validation EventHandler nil
async throws
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer
where InputSequence Sequence Validation Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Base Annotation
Validation Element AnnotatedFeature FeatureSequence
Base Annotation FeatureSequence Feature

```

Base Transformer Input

```
/// Updates a transformer with a new sequence of examples.  
///  
/// - Parameters:  
///   - transformer: A transformer to update.  
///   - input: A sequence of examples.  
///   - eventHandler: An event handler.  
@inlinable public func update InputSequence  
FeatureSequence _ inout  
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer  
InputSequence EventHandler  
nil async throws where InputSequence Sequence  
FeatureSequence TemporalSequence InputSequence Element  
AnnotatedFeature FeatureSequence Base Annotation  
FeatureSequence Feature Base Transformer Input  
  
/// Encodes a fitted transformer.  
@inlinable public func encode _  
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer  
inout any EstimatorEncoder throws  
  
/// Decodes the transformer.  
@inlinable public func decode inout any  
EstimatorDecoder throws  
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer  
  
/// Encodes the transformer and optimizer to an encoder.  
///  
/// - Parameters:  
///   - transformer: A transformer this estimator creates.  
///   - encoder: An encoder.  
public func encodeWithOptimizer _  
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer  
inout any EstimatorEncoder throws  
  
/// Reads the encoded transformer and optimizer with a decoder.  
///  
/// - Parameter decoder: A decoder.  
/// - Returns: The decoded transformer.  
public func decodeWithOptimizer inout any  
EstimatorDecoder throws  
UpdatableSupervisedEstimatorToTemporalAdaptor Base Transformer
```

@available	13.0	15.0
@available	16.0	18.0

```

@available(16.0, 18.0)
@available(1.0, 2.0)
@available
extension UpdatableSupervisedEstimatorToTemporalAdaptor
Sendable where Base: Sendable

/// A supervised tabular estimator that can be incrementally updated.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public protocol UpdatableSupervisedTabularEstimator : SupervisedTabularEstimator

/// Creates a default-initialized transformer suitable for incremental fitting.
func makeTransformer() Self Transformer

/// Updates a transformer with a new sequence of examples.
///
/// - Parameters:
///   - transformer: A transformer to update.
///   - input: A data frame containing examples.
///   - eventHandler: An event handler.
func update(_ transformer: inout Self Transformer, input: DataFrame, eventHandler: EventHandler) async throws

/// Encodes the transformer and optimizer to an encoder.
///
/// - Parameters:
///   - transformer: A transformer this estimator creates.
///   - encoder: An encoder.
func encodeWithOptimizer(_ transformer: inout any EstimatorEncoder) throws

/// Reads the encoded transformer and optimizer with a decoder.
///
/// - Parameter decoder: A decoder.
/// - Returns: The decoded transformer.
func decodeWithOptimizer(decoder: EstimatorDecoder) throws Self Transformer

extension UpdatableSupervisedTabularEstimator : ComposedTabularTransformer

/// Composes this supervised tabular estimator with a tabular transformer.
@available(macOS 13.0, iOS 16.0, tvOS 16.0, watchOS 11.0)
public func appending<Other: Self Transformer>(_ other: Other) Some<UpdatableSupervisedTabularEstimator<Other>>

```

```
TabularTransformer Self Annotation Equatable
```

```
    /// Composes this supervised tabular estimator with an updatable tabular
estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func appending Other -> Other some
UpdatableSupervisedTabularEstimator ComposedTabularTransformer
Self Transformer Other Transformer Self Annotation where
Other UpdatableTabularEstimator Self Annotation Equatable
```

```
    /// Composes this supervised tabular estimator with another supervised
tabular estimator.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func appending Other -> Other some
UpdatableSupervisedTabularEstimator ComposedTabularTransformer
Self Transformer Other Transformer Self Annotation where
Other UpdatableSupervisedTabularEstimator Self Annotation
Other Annotation
```

```
extension UpdatableSupervisedTabularEstimator
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func update _> inout
Self Transformer DataFrame async throws
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension UpdatableSupervisedTabularEstimator
```

```
    /// Writes the encoded transformer and optimizer to a file.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - url: A file URL.
    ///   - overwrite: A Boolean value indicating whether to overwrite
existing files.
    public func writeWithOptimizer _>
Self Transformer URL Bool true throws
```

```
    /// Reads the encoded transformer and optimizer from a file.
    ///
    /// - Parameter url: A file URL.
    /// - Returns: The decoded transformer.
```

```

public func readWithOptimizer throws
Self Transformer URL throws

/// A supervised temporal estimator that can be incrementally updated.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public protocol
UpdatableSupervisedTemporalEstimator
SupervisedTemporalEstimator

/// Creates a default-initialized transformer suitable for incremental fitting.
func makeTransformer Self Transformer

/// Updates a transformer with a new sequence of examples.
///
/// - Parameters:
///   - transformer: A transformer to update.
///   - input: A sequence of examples.
///   - eventHandler: An event handler.
func update InputSequence FeatureSequence -
inout Self Transformer InputSequence
EventHandler async throws where
InputSequence Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Self Annotation FeatureSequence Feature
Self Transformer Input

/// Encodes the transformer and optimizer to an encoder.
///
/// - Parameters:
///   - transformer: A transformer this estimator creates.
///   - encoder: An encoder.
func encodeWithOptimizer Self Transformer
inout any EstimatorEncoder throws

/// Reads the encoded transformer and optimizer with a decoder.
///
/// - Parameter decoder: A decoder.
/// - Returns: The decoded transformer.
func decodeWithOptimizer inout any
EstimatorDecoder throws Self Transformer

@available 13.0 15.0
@available 16.0 18.0

```

```

@available 16.0 18.0
@available 1.0 2.0
@available
extension UpdatableSupervisedTemporalEstimator

    /// Writes the encoded transformer and optimizer to a file.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - url: A file URL.
    ///   - overwrite: A Boolean value indicating whether to overwrite
existing files.
    public func writeWithOptimizer _  

Self Transformer URL Bool true throws

    /// Reads the encoded transformer and optimizer from a file.
    ///
    /// - Parameter url: A file URL.
    /// - Returns: The decoded transformer.
    public func readWithOptimizer URL throws  

Self Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension UpdatableSupervisedTemporalEstimator

    /// Composes this updatable supervised temporal estimator with a
transformer.
    public func appending Other _  

UpdatableSupervisedTemporalEstimator ComposedTemporalTransform  

er Self Transformer TransformerToTemporalAdaptor Other  

Self Annotation where Other Transformer Other Input  

Self Transformer Output

    /// Composes this updatable supervised temporal estimator with an
updatable estimator.
    public func appending Other _  

UpdatableSupervisedTemporalEstimator ComposedTemporalTransform  

er Self Transformer  

TransformerToTemporalAdaptor Other Transformer  

Self Annotation where Other UpdatableEstimator  

Self Transformer Output Other Transformer Input

    /// Composes this updatable supervised temporal estimator with an

```

updatable supervised estimator.

```
public func appending Other _          Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer
TransformerToTemporalAdaptor Other Transformer
Self Annotation where Other UpdatableSupervisedEstimator
Self Annotation      Other Annotation Self Transformer Output
Other Transformer Input
```

/// Composes this updatable supervised temporal estimator with a transformer.

```
public func appending Other _          Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer Other Self Annotation where Other TemporalTransformer Other Input Self Transformer Output
```

/// Composes this updatable supervised temporal estimator with an updatable temporal estimator.

```
public func appending Other _          Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer Other Transformer Self Annotation
where Other UpdatableTemporalEstimator
Self Transformer Output Other Transformer Input
```

/// Composes this updatable supervised temporal estimator with another updatable supervised temporal estimator.

```
public func appending Other _          Other      some
UpdatableSupervisedTemporalEstimator ComposedTemporalTransformer Self Transformer Other Transformer Self Annotation
where Other UpdatableSupervisedTemporalEstimator
Self Annotation      Other Annotation Self Transformer Output
Other Transformer Input
```

@available	13.0	15.0
@available	16.0	18.0
@available	16.0	18.0
@available	1.0	2.0
@available		
extension UpdatableSupervisedTemporalEstimator		

```
    @inlinable public func update InputSequence
FeatureSequence _           inout Self Transformer
InputSequence  async throws where InputSequence
Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
```

```
Self Annotation FeatureSequence Feature
Self Transformer Input
```

```
/// A tabular estimator that can be incrementally updated.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public protocol UpdatableTabularEstimator
    TabularEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
    func makeTransformer           Self Transformer

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A data frame containing examples.
    ///   - eventHandler: An event handler.
    func update _                  inout Self Transformer
        DataFrame                EventHandler    async throws

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
    func encodeWithOptimizer _     inout any EstimatorEncoder throws
        Self Transformer

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    func decodeWithOptimizer      inout any
        EstimatorDecoder    throws    Self Transformer

extension UpdatableTabularEstimator

    /// Exposes this updatable tabular estimator as a supervised tabular
    estimator.
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
    public func
    adaptedAsSupervised Annotation
    ColumnID Annotation
    UpdatableTabularEstimatorToSupervisedAdaptor Self Annotation
    where Annotation Equatable
```

```
extension UpdatableTabularEstimator

    /// Composes this updatable tabular estimator with an updatable supervised
tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other some
UpdatableSupervisedTabularEstimator ComposedTabularTransformer
    Self Transformer Other Transformer Other Annotation where
Other UpdatableSupervisedTabularEstimator
Other Annotation Equatable
```

```
extension UpdatableTabularEstimator

    /// Composes this updatable tabular estimator with a tabular transformer.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other some
UpdatableTabularEstimator ComposedTabularTransformer Self Tran
sformer Other where Other TabularTransformer
```

```
    /// Composes this updatable tabular estimator with another updatable tabular
estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func appending Other _ Other some
UpdatableTabularEstimator ComposedTabularTransformer Self Tran
sformer Other Transformer where Other
UpdatableTabularEstimator
```

```
extension UpdatableTabularEstimator

    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

        @inlinable public func update _ inout
Self Transformer DataFrame async throws
```

```
    /// An adaptor that exposes an updatable tabular estimator as an updatable
supervised tabular estimator.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
public struct
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation UpdatableSupervisedTabularEstimator where
```

```
Estimator UpdatableTabularEstimator Annotation Equatable

    /// The transformer type created by this estimator.
public typealias Transformer Estimator Transformer

    /// The annotation column identifier.
public var annotationColumnID ColumnID Annotation

    /// The wrapped estimator.
public let estimator Estimator

    /// Creates an updatable tabular estimator supervised adaptor.
public init _ Estimator
ColumnID Annotation

    /// Creates a default-initialized transformer suitable for incremental fitting.
public func makeTransformer Estimator Transformer

    /// Fits a transformer to a data frame.
    ///
    /// - Parameters:
    ///   - input: A data frame containing examples used for fitting the
    transformer.
    ///   - validation: A data frame containing examples used for
    validating the fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
public func fitted DataFrame
DataFrame EventHandler nil
async throws
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer

    /// Updates a transformer with a new data frame containing examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
public func update _ inout
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer DataFrame
EventHandler nil async throws

    /// Does nothing since this estimator uses a pre-defined transformer.
public func encode _
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer inout any
EstimatorEncoder throws
```

```

    /// Returns the pre-defined transformer.
    public func decode inout any
EstimatorDecoder throws
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer created by this estimator.
    ///   - encoder: An encoder.
    public func encodeWithOptimizer UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer inout any
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    public func decodeWithOptimizer inout any
EstimatorDecoder throws
UpdatableTabularEstimatorToSupervisedAdaptor Estimator
Annotation Transformer

```

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension UpdatableTabularEstimatorToSupervisedAdaptor
Sendable where Estimator Sendable Annotation Sendable

```

    /// A temporal estimator that can be incrementally updated.
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
public protocol UpdatableTemporalEstimator
TemporalEstimator

    /// Creates a default-initialized transformer suitable for incremental fitting.
func makeTransformer Self Transformer

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.

```

```

    /// - eventHandler: An event handler.
    func update InputSequence _ inout
Self Transformer Input Sequence
Event Handler async throws where InputSequence Sequence
InputSequence Element TemporalSequence
Self Transformer Input InputSequence Element Feature

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    /// - transformer: A transformer this estimator creates.
    /// - encoder: An encoder.
    func encodeWithOptimizer _ Self Transformer
        inout any EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
    func decodeWithOptimizer inout any
EstimatorDecoder throws Self Transformer

@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension UpdatableTemporalEstimator

    /// Composes this updatable temporal estimator with a transformer.
    public func appending Other _ Other some
UpdatableTemporalEstimator ComposedTemporalTransformer Self Transformer
Transformer TransformerToTemporalAdaptor Other where Other Transformer
Transformer Other Input Self Transformer Output

    /// Composes this updatable temporal estimator with an updatable estimator.
    public func appending Other _ Other some
UpdatableTemporalEstimator ComposedTemporalTransformer Self Transformer
Transformer TransformerToTemporalAdaptor Other Transformer
where Other UpdatableEstimator Self Transformer Output
Other Transformer Input

    /// Composes this updatable temporal estimator with a temporal transformer.
    public func appending Other _ Other some
UpdatableTemporalEstimator ComposedTemporalTransformer Self Transformer
Other where Other TemporalTransformer

```

Other Input Self Transformer Output

```
    /// Composes this updatable temporal estimator with another updatable
    temporal estimator.
    public func appending Other _           Other      some
UpdatableTemporalEstimator ComposedTemporalTransformer Self Tr
ansformer Other Transformer where Other
UpdatableTemporalEstimator Self Transformer Output
Other Transformer Input
```

```
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
extension UpdatableTemporalEstimator
```

```
    /// Exposes this temporal estimator as a supervised temporal estimator.
    public func
adaptedAsSupervised Annotation
Annotation           self
UpdatableTemporalEstimatorToSupervisedAdaptor Self
Annotation where Annotation Equatable Annotation
Sendable
```

```
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
extension UpdatableTemporalEstimator
```

```
    @inlinable public func update InputSequence _
        inout Self Transformer
InputSequence  async throws where InputSequence Sequence
InputSequence Element TemporalSequence
Self Transformer Input InputSequence Element Feature
```

```
@available           13.0          15.0
@available           16.0          18.0
@available           16.0          18.0
@available           1.0           2.0
@available
extension UpdatableTemporalEstimator
```

```
    /// Composes this updatable temporal estimator with an updatable  
    supervised temporal estimator.  
    @preconcurrency public func appending Other _  
    Other some  
UpdatableSupervisedTemporalEstimator ComposedTemporalTransform  
er Self Transformer  
TransformerToTemporalAdaptor Other Transformer  
Other Annotation where Other UpdatableSupervisedEstimator  
Other Annotation Sendable Self Transformer Output  
Other Transformer Input
```

```
    /// Composes this updatable temporal estimator with an updatable  
    supervised temporal estimator.  
    public func appending Other _ Other some  
UpdatableSupervisedTemporalEstimator ComposedTemporalTransform  
er Self Transformer Other Transformer Other Annotation  
where Other UpdatableSupervisedTemporalEstimator  
Self Transformer Output Other Transformer Input
```

```
/// An adaptor that exposes an updatable temporal estimator as an updatable  
supervised temporal estimator.
```

```
@available 13.0 15.0  
@available 16.0 18.0  
@available 16.0 18.0  
@available 1.0 2.0  
@available  
public struct  
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator  
Annotation UpdatableSupervisedTemporalEstimator where  
Estimator UpdatableTemporalEstimator Annotation  
Equatable Annotation Sendable
```

```
/// The transformer type created by this estimator.
```

```
public typealias Transformer Estimator Transformer
```

```
/// The wrapped estimator.
```

```
public let estimator Estimator
```

```
/// Creates a temporal estimator adaptor.
```

```
public init _ Estimator
```

```
/// Creates a default-initialized transformer suitable for incremental fitting.  
public func makeTransformer Estimator Transformer
```

```
/// Fits a transformer to a sequence of examples.
```

```
///
```

```

    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted InputSequence FeatureSequence
      InputSequence           EventHandler     nil async
throws
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where InputSequence Sequence
FeatureSequence TemporalSequence InputSequence Element
AnnotatedFeature FeatureSequence Annotation
FeatureSequence Feature Estimator Transformer Input

    /// Fits a transformer to a sequence of examples while validating with a
validation sequence.
    ///
    /// - Parameters:
    ///   - input: A sequence of examples used for fitting the transformer.
    ///   - validation: A sequence of examples used for validating the
fitted transformer.
    ///   - eventHandler: An event handler.
    /// - Returns: The fitted transformer.
    public func fitted InputSequence Validation
      FeatureSequence           InputSequence
                                Validation           EventHandler     nil
async throws
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer where InputSequence Sequence
Validation Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Annotation Validation Element
AnnotatedFeature FeatureSequence Annotation
FeatureSequence Feature Estimator Transformer Input

    /// Updates a transformer with a new sequence of examples.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - eventHandler: An event handler.
    public func update InputSequence FeatureSequence _
      inout
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer           InputSequence
                           EventHandler     nil async throws where
InputSequence Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Annotation FeatureSequence Feature
Estimator Transformer Input

```

```

    /// Fits a transformer to a sequence of examples while validating with a
validation sequence.
    ///
    /// - Parameters:
    ///   - transformer: A transformer to update.
    ///   - input: A sequence of examples.
    ///   - validation: A sequence of examples used for validation.
    ///   - eventHandler: An event handler.
public func update InputSequence Validation
FeatureSequence inout
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer InputSequence
Validation EventHandler nil
async throws where InputSequence Sequence Validation
Sequence FeatureSequence TemporalSequence
InputSequence Element AnnotatedFeature FeatureSequence
Annotation Validation Element
AnnotatedFeature FeatureSequence Annotation
FeatureSequence Feature Estimator Transformer Input

    /// Does nothing since this estimator uses a pre-defined transformer.
public func encode _
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer inout any
EstimatorEncoder throws

    /// Returns the pre-defined transformer.
public func decode inout any
EstimatorDecoder throws
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer

    /// Encodes the transformer and optimizer to an encoder.
    ///
    /// - Parameters:
    ///   - transformer: A transformer this estimator creates.
    ///   - encoder: An encoder.
public func encodeWithOptimizer _
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator
Annotation Transformer inout any
EstimatorEncoder throws

    /// Reads the encoded transformer and optimizer with a decoder.
    ///
    /// - Parameter decoder: A decoder.
    /// - Returns: The decoded transformer.
public func decodeWithOptimizer inout any
EstimatorDecoder throws
UpdatableTemporalEstimatorToSupervisedAdaptor Estimator

```

Annotation Transformer

```
@available 13.0 15.0
@available 16.0 18.0
@available 16.0 18.0
@available 1.0 2.0
@available
extension UpdatableTemporalEstimatorToSupervisedAdaptor
Sendable where Estimator Sendable

/// An async sequence of augmented elements.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct UpsampledAugmentationSequence Base
RandomTransformer RandomNumberGenerator Annotation
AsyncSequence where Base Collection RandomTransformer
RandomTransformer RandomNumberGenerator
RandomNumberGenerator Base Element
AnnotatedFeature RandomTransformer Input Annotation
RandomTransformer Input RandomTransformer Output

/// The type of element produced by this asynchronous sequence.
public typealias Element Base Element

/// The transformation applied to each element.
public let transformer RandomTransformer

/// Creates the asynchronous iterator that produces elements of this
asynchronous sequence.
public func makeAsyncIterator
UpsampledAugmentationSequence Base RandomTransformer
RandomNumberGenerator Annotation AsyncIterator

extension UpsampledAugmentationSequence

/// The iterator that produces elements in the augmentation sequence.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
public struct AsyncIterator AsyncIteratorProtocol

/// Produces the next element in the augmentation sequence.
public mutating func next async throws
Base Element

@available iOS 17.0 tvOS 17.0 watchOS 11.0 macOS
14.0
public typealias Element
```

```
AnnotatedFeature RandomTransformer Input Annotation
```

```
/// A video file reader.  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public struct VideoReader : Transformer, Sendable  
  
/// Creates a video reader.  
public init  
  
/// Reads a video file as an async sequence of video frames.  
///  
/// - Parameters:  
///   - url: A video file URL.  
///   - eventHandler: An event handler.  
/// - Returns: An async sequence of `VideoFrames`.  
public func applied URL  
EventHandler async throws VideoReader AsyncFrames  
  
/// Reads a video file as an async sequence of video frames.  
///  
/// - Parameter url: A video file URL.  
/// - Returns: An async sequence of `VideoFrames`.  
public static func read URL async throws  
VideoReader AsyncFrames  
  
/// Reads a sequence of files as an array of async sequences of video  
frames.  
///  
/// - Parameter files: A sequence of URLs.  
/// - Returns: An array of async sequences of video frames.  
public static func read S _ S async throws  
VideoReader AsyncFrames where S : Sequence, S.Element : URL  
  
/// Reads a sequence of annotated files as an array of annotated async  
sequences of video frames.  
///  
/// - Parameter annotatedFiles: A sequence of annotated URLs.  
/// - Returns: An array of annotated async sequences.  
public static func read S Annotation _  
S async throws AnnotatedFeature VideoReader AsyncFrames  
Annotation where S : Sequence, Annotation : Equatable  
Annotation : Sendable, S.Element : AnnotatedFeature, URL  
Annotation  
  
/// Reads an async sequence of video frames captured with a video camera.  
///  
/// - Parameter configuration: The camera configuration  
/// - Returns: An async sequence of camera video frames.
```

```


@available(macOS 13.0, iOS 16.0)
@available
public static func readCamera(VideoReader CameraConfiguration) async throws VideoReader CameraAsyncBuffers

    /// The input type.
@available(iOS 16.0, tvOS 16.0, macOS 13.0)
public typealias Input = URL

    /// The output type.
@available(iOS 16.0, tvOS 16.0, macOS 13.0)
public typealias Output = VideoReader<AsyncFrames>

extension VideoReader {
    /// The configuration of the camera to pass to the `readCamera` method.
@available(macOS 13.0, iOS 16.0)
@available
public struct CameraConfiguration : Sendable {
        /// The position of the camera for an iOS device.
public enum Position : Sendable {
            case front
            case back
        }

        /// Returns a Boolean value indicating whether two values are equal.
        /**
         * Equality is the inverse of inequality. For any values `a` and `b`,
         * `a == b` implies that `a != b` is `false`.
         */
        /// - Parameters:
        ///   - lhs: A value to compare.
        ///   - rhs: Another value to compare.
        public static func ==(lhs: CameraConfiguration, rhs: CameraConfiguration) Bool {
            /**
             * Hashes the essential components of this value by feeding them into the
             * given hasher.
             */
            /**
             * Implement this method to conform to the `Hashable` protocol. The
             * components used for hashing must be the same as the
             */
        }
    }
}


```

```
components compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,
    /// don't call `finalize()` on the `hasher` instance
provided,
    /// or replace it with a different instance.
    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining
the components
    /// of this instance.
public func hash           inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a
`Hashable` requirement. To
    /// conform to `Hashable`, implement the `hash(into:)` requirement instead.
    /// The compiler provides an implementation for `hashValue` for you.
public var hashValue  Int   get

    /// The camera pixel format.
public enum PixelFormat  Sendable

    case bgra32

    case yCbCr8BiPlanarFullRange420

    /// Returns a Boolean value indicating whether two values are
equal.
    ///
    /// Equality is the inverse of inequality. For any values `a` and
`b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    /// - lhs: A value to compare.
    /// - rhs: Another value to compare.
public static func
```

```
VideoReader CameraConfiguration PixelFormat
VideoReader CameraConfiguration PixelFormat      Bool

    /// Hashes the essential components of this value by feeding
them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable`  

protocol. The
    /// components used for hashing must be the same as the
components compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`  

    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,  

    /// don't call `finalize()` on the `hasher` instance  

provided,  

    /// or replace it with a different instance.  

    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining
the components
    /// of this instance.
public func hash           inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a
`Hashable` requirement. To
    /// conform to `Hashable`, implement the `hash(into:)`  

requirement instead.
    /// The compiler provides an implementation for `hashValue`  

for you.
public var hashValue  Int   get

    /// The camera resolution.
public enum Resolution  Sendable

    case low
    case medium
    case high
```

```

    /// Returns a Boolean value indicating whether two values are
equal.

    ///
    /// Equality is the inverse of inequality. For any values `a` and
`b`,
    /// `a == b` implies that `a != b` is `false`.
    ///
    /// - Parameters:
    ///   - lhs: A value to compare.
    ///   - rhs: Another value to compare.
public static func
VideoReader CameraConfiguration Resolution
VideoReader CameraConfiguration Resolution Bool

    /// Hashes the essential components of this value by feeding
them into the
    /// given hasher.
    ///
    /// Implement this method to conform to the `Hashable`
protocol. The
    /// components used for hashing must be the same as the
components compared
    /// in your type's `==` operator implementation. Call
`hasher.combine(_:)`
    /// with each of these components.
    ///
    /// - Important: In your implementation of `hash(into:)`,  

    /// don't call `finalize()` on the `hasher` instance
provided,
    /// or replace it with a different instance.  

    /// Doing so may become a compile-time error in the future.
    ///
    /// - Parameter hasher: The hasher to use when combining
the components
    /// of this instance.
public func hash inout Hasher

    /// The hash value.
    ///
    /// Hash values are not guaranteed to be equal across different
executions of
    /// your program. Do not save hash values to use during a future
execution.
    ///
    /// - Important: `hashValue` is deprecated as a
`Hashable` requirement. To
        /// conform to `Hashable`, implement the `hash(into:)`  

requirement instead.
    /// The compiler provides an implementation for `hashValue`  

for you.
public var hashValue Int get

```

```

    /// The camera position. For an iOS device this can be either
`front` or `rear`.
    /// For devices with just one camera this value is ignored.
    /// The default value is `front`.
@available          13.0           14.0
    "Use the position property instead."
@available          16.0           17.0
    "Use the position property instead."
@available
public var cameraPosition
VideoReader CameraConfiguration Position get

    /// The camera position. For an iOS device this can be either
`front` or `rear`.
    /// For devices with just one camera this value is ignored.
    /// The default value is `front`.
public var position
VideoReader CameraConfiguration Position

    /// The camera pixel format. The default is `bgra32`.
public var pixelFormat
VideoReader CameraConfiguration PixelFormat

    /// The camera resolution specifying the quality of the video output. The
default values is `high`
public var resolution
VideoReader CameraConfiguration Resolution

    /// The camera frame rate. The default value is 30.0 frames per
second.
public var frameRate Double

    /// Creates a camera configuration.
public init

    /// Creates a camera configuration.
    /// - Parameters:
    ///   - position: The position of the camera. The default value is
`front`. For devices with just one camera this value is ignored.
    ///   - pixelFormat: The pixel format of the camera frames. The
default is `bgra32`.
    ///   - resolution: The camera resolution. The default values is
`high`.
    ///   - frameRate: The camera frame rate. The default value is
30.0 frames per second.
public init
VideoReader CameraConfiguration Position
    VideoReader CameraConfiguration PixelFormat

```

```
VideoReader CameraConfiguration Resolution  
Double 30.0
```

```
extension VideoReader
```

```
    /// An async sequence of video frames.  
    ///  
    /// This sequence allows iterating through the file only once.  
    @available(macOS 13.0, iOS 16.0, tvOS 16.0)  
    public struct AsyncFrames : TemporalSequence
```

```
        /// The type of asynchronous iterator that produces elements of this  
        /// asynchronous sequence.  
        public typealias AsyncIterator  
VideoReader AsyncFrames Iterator
```

```
        /// The feature type.  
        public typealias Feature = CIImage
```

```
        /// The number of frames. For this sequence count is always nil.  
        public var count: Int { get }
```

```
        /// The video file URL, used when throwing an error.  
        public let url: URL
```

```
        /// The timescale of the video track.  
        public let timescale: CMTimeScale
```

```
        /// The nominal frame rate.  
        public let nominalFrameRate: Float
```

```
        /// The frame size.  
        public let frameSize: CGSize
```

```
        /// The video duration.  
        public let videoDuration: CMTime
```

```
        /// Constructs an iterator.  
        public func makeAsyncIterator  
VideoReader AsyncFrames Iterator
```

```
        /// The type of element produced by this asynchronous sequence.  
        @available(iOS 16.0, tvOS 16.0, macOS 13.0)  
        public typealias Element  
TemporalFeature VideoReader AsyncFrames Feature
```

```
extension VideoReader
```

```
    /// An async sequence of video frames.  
    ///  
    /// This sequence allows iterating through the camera frames. Only one  
    iterator can be created per sequence.  
    @available macOS 13.0 iOS 16.0  
    @available  
    public struct CameraAsyncBuffers : TemporalSequence  
  
        /// The type of asynchronous iterator that produces elements of this  
        /// asynchronous sequence.  
        public typealias AsyncIterator  
VideoReader CameraAsyncBuffers Iterator  
  
        /// The feature type.  
        public typealias Feature = CIImage  
  
        /// The number of frames. For this sequence count is always nil.  
        public var count : Int  
        get  
  
            /// The capture session.  
            ///  
            /// You can use the capture session to create a preview with  
            ///  
<doc://com.apple.documentation/documentation/avfoundation/avcapturevideoprevie  
wlayer> and to configure the  
    /// input device, for example switching the input camera.  
    ///  
    /// ``  
    /// let sequence = try await  
VideoReader.readCamera(configuration: configuration)  
    /// sequence.captureSession.beginConfiguration()  
    ///  
sequence.captureSession.removeInput(captureSession.inputs[0])  
    /// try  
sequence.captureSession.addInput( AVCaptureDeviceInput(device:  
camera))  
    /// sequence.captureSession.commitConfiguration()  
    /// ``  
    @available macOS 15.0 iOS 18.0  
    @available  
    public var captureSession : AVCaptureSession  
    get  
  
        /// Constructs an iterator.  
        public consuming func makeAsyncIterator  
VideoReader CameraAsyncBuffers Iterator
```

```
    /// The type of element produced by this asynchronous sequence.  
    @available iOS 16.0 macOS 13.0  
    @available  
        public typealias Element  
    TemporalFeature VideoReader CameraAsyncBuffers Feature
```

```
extension VideoReader AsyncFrames
```

```
    /// An async iterator of video frames.  
    @available macOS 13.0 iOS 16.0 tvOS 16.0  
    public struct Iterator : AsyncIteratorProtocol
```

```
        /// Asynchronously advances to the next element and returns it, or  
        ends the  
            /// sequence if there is no next element.  
            ///  
            /// - Returns: The next element, if it exists, or `nil` to signal the  
        end of  
            /// the sequence.  
        public mutating func next() async throws  
    TemporalFeature CIImage
```

```
    @available iOS 16.0 tvOS 16.0 macOS 13.0  
    public typealias Element = TemporalFeature CIImage
```

```
@available macOS 15.0 iOS 18.0  
@available  
extension VideoReader CameraAsyncBuffers : Sendable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraAsyncBuffers
```

```
    /// An async iterator of video frames.  
    @available macOS 13.0 iOS 16.0  
    @available  
        final public class Iterator : AsyncIteratorProtocol
```

```
            /// Advances to the next element and returns it, or nil if no next element  
            exists.  
            final public func next() async throws  
    TemporalFeature CIImage
```

```
    @available iOS 16.0 macOS 13.0
```

```
@available public typealias Element TemporalFeature CIImage 16.0
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration Position Equatable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration Position Hashable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration PixelFormat Equatable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration PixelFormat Hashable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration Resolution Equatable
```

```
@available macOS 13.0 iOS 16.0  
@available  
extension VideoReader CameraConfiguration Resolution Hashable
```

```
/// Video loader errors.
```

```
@available macOS 13.0 iOS 16.0 tvOS 16.0  
public enum VideoReaderError LocalizedError Equatable Sendable
```

```
/// An error that indicates that the VideoReader cannot find a video track.  
case missingVideoTrack URL
```

```
    /// An error that indicates that the camera authorization status is denied.  
    /// The user has explicitly denied permission for media capture.  
    case cameraAuthorizationDenied  
  
    /// An error that indicates that the camera authorization status is restricted.  
    /// The user is not allowed to access media capture devices.  
    case cameraAuthorizationRestricted  
  
    /// An error that indicates that no cameras are available.  
    case sourceCameraNotAvailable  
  
    /// An error that indicates that the frame rate is not supported by the input  
camera.  
    case frameRateNotSupported Double  
  
    /// An error that indicates that the capture session stopped.  
@available macOS 15.0 iOS 18.0 tvOS 18.0  
    case captureSessionStopped  
  
    /// A localized message describing what error occurred.  
public var errorMessage String get  
  
    /// Returns a Boolean value indicating whether two values are equal.  
    ///  
    /// Equality is the inverse of inequality. For any values `a` and `b`,  
    /// `a == b` implies that `a != b` is `false`.  
    ///  
    /// - Parameters:  
    ///   - lhs: A value to compare.  
    ///   - rhs: Another value to compare.  
public static func VideoReaderError  
VideoReaderError Bool  
  
@available macOS 13.0 iOS 16.0 tvOS 16.0  
extension VideoReaderError CustomDebugStringConvertible  
  
    /// A text representation of the error.  
public var debugDescription String get  
  
    /// Computes the maximum absolute error between predicted and ground truth  
values.  
    ///  
    /// Empty collections of predicted and ground truth values will return a value of  
NaN.  
    ///  
    /// - Parameters:  
    ///   - predicted: The predicted values.  
    ///   - groundTruth: The ground truth values. The collection must have same
```

```

number of elements as the predicted values.

/// - Returns: The maximum absolute error.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
@backDeployed(before: macOS 14.0, iOS 17.0, tvOS 17.0)
@inlinable public func maximumAbsoluteError T -
some Collection _ some Collection T where
T FloatingPoint

/// Computes the maximum absolute error between predicted and ground truth
values.

///
/// If an empty `AnnotatedPrediction` is supplied, the result will be NaN.

/// - Parameters:
///   - annotatedPredictions: An `AnnotatedPrediction` object.
/// - Returns: The maximum absolute error.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
@inlinable public func maximumAbsoluteError T -
AnnotatedPrediction T T T where
T FloatingPoint

/// Computes the mean absolute error between predicted and ground truth values.

///
/// Empty collections of predicted and ground truth values will return a value of
NaN.

///
/// - Parameters:
///   - predicted: The predicted values.
///   - groundTruth: The ground truth values. The collection must have same
number of elements as the predicted values.
/// - Returns: The mean absolute error.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
@backDeployed(before: macOS 14.0, iOS 17.0, tvOS 17.0)
@inlinable public func meanAbsoluteError T -
some Collection _ some Collection T where T
FloatingPoint

/// Computes the mean absolute error between predicted and ground truth values.

///
/// If an empty `AnnotatedPrediction` is supplied, the result will be NaN.

/// - Parameters:
///   - annotatedPredictions: An `AnnotatedPrediction` object.
/// - Returns: The mean absolute error.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
@inlinable public func meanAbsoluteError T -
AnnotatedPrediction T T T where
T FloatingPoint

/// Computes the mean absolute percentage error between predicted and ground

```

```

truth values.

///
/// If an empty `AnnotatedPrediction` is supplied, the result will be NaN.
///
/// - Parameters:
///   - annotatedPredictions: An `AnnotatedPrediction` object.
/// - Returns: The mean absolute percentage error as a decimal value.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
@inlinable public func meanAbsolutePercentageError T
                           AnnotatedPrediction T T      T where
T FloatingPoint

/// Computes the mean squared error between predicted and ground truth values.
///
/// Empty collections of predicted and ground truth values will return a value of
NaN.
///
/// - Parameters:
///   - predicted: The predicted values.
///   - groundTruth: The ground truth values. The collection must have same
number of elements as the predicted values.
/// - Returns: The mean squared error.
@available macOS 13.0 iOS 16.0 tvOS 16.0 visionOS 1.0
watchOS 11.0
@backDeployed(before: macOS 15.0, iOS 18.0, tvOS 18.0,
visionOS 2.0)
@inlinable public func meanSquaredError T
                           some
Collection _           some Collection      T where T
FloatingPoint

/// Computes the root mean squared error between predicted and ground truth
values.
///
/// If an empty `AnnotatedPrediction` is supplied, the result will be NaN.
///
/// - Parameters:
///   - annotatedPredictions: An `AnnotatedPrediction` object.
/// - Returns: The root mean squared error.
@available macOS 14.0 iOS 17.0 tvOS 17.0 visionOS 1.0
watchOS 11.0
@backDeployed(before: macOS 15.0, iOS 18.0, tvOS 18.0,
visionOS 2.0)
@inlinable public func meanSquaredError T
                           AnnotatedPrediction T T      T where
T FloatingPoint

/// Computes the root mean squared error between predicted and ground truth
values.
///
/// Empty collections of predicted and ground truth values will return a value of

```

```

NaN.
///
/// - Parameters:
///   - predicted: The predicted values.
///   - groundTruth: The ground truth values. The collection must have same
///     number of elements as the predicted values.
/// - Returns: The root mean squared error.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
@backDeployed(before: macOS 14.0, iOS 17.0, tvOS 17.0)
@inlinable public func rootMeanSquaredError T _
some Collection T some Collection T where
T FloatingPoint

/// Computes the root mean squared error between predicted and ground truth
values.
///
/// If an empty `AnnotatedPrediction` is supplied, the result will be NaN.
///
/// - Parameters:
///   - annotatedPredictions: An `AnnotatedPrediction` object.
/// - Returns: The root mean squared error.
@available macOS 14.0 iOS 17.0 tvOS 17.0 watchOS 11.0
@inlinable public func rootMeanSquaredError T _
AnnotatedPrediction T T where
T FloatingPoint

extension DataFrame

    /// Generates a data frame that includes only the column selection.
    ///
    /// - Parameter selection: A selection of columns.
    /// - Returns: A new data frame including only the selected columns.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

    public func selecting _ ColumnSelection
DataFrame

@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
extension DataFrame

    /// Loads training examples from a data frame containing annotations.
    ///
    /// The exact column ids within the data frame which contains the file path,
    annotation, startTime
        /// and endTime of each temporal annotation, can be customized using the
        `parameters`
            /// argument.
            ///
    /// The data frame must contain at least two columns: one for file path and

```

one for annotation.

```
    /// The file path must be of String type and annotation can be of any
    equatable type.
    ///
    /// The `filePath` column in the `dataFrame` should contain file path
    with its extension. All the files
    /// can have absolute path or a path relative to a `baseURL`. This option
    can be specified using
    /// `filePathType` in `parameters`.
    ///
    /// The `startTime` and `endTime` columns contain the start time and
    end time of a temporal segment
    /// in seconds. Their type is Double. Both the columns are optional.
    ///
    /// If only a `startTime` is specified, it is interpreted as an annotation for
    the range starting from
    /// `startTime` till the end of file, i.e., `startTime ...`.
    ///
    /// If only an `endTime` is specified, it is interpreted as an annotation for
    the range starting at 0 till
    /// `endTime`, i.e., `0 ..< endTime`.
    ///
    /// If both `startTime` and `endTime` columns are missing or their
    values are missing, the annotation
    /// is assumed to be for the entire file length.
    ///
    /// – Note: Ensure that the same file has no overlapping range in the
    dataframe. Since overlapping
    /// ranges would mean conflicting annotation for that range.
    ///
    /// Consider an example of a dataframe containing the following values:
    ///
    ///
    //|-----|
    //| fileName | annotation | startTime | endTime |
    //|-----|
    //| file1.wav | class1 | 0 | 4 |
    //| file1.wav | class2 | 5 | 10 |
    //| file2.wav | class1 | 2 | null |
    //| file3.wav | class2 | null | null |
    //| file4.wav | class1 | null | 5 |
    //|-----|
    ///
    ///
    /// This will create an annotated feature list which has the following structure:
    /// ````swift
    /// [
    ///     AnnotatedFeature(feature:
    TemporalFileSegment("file1.wav", 0 ..< 4), annotation:
    "class1"),
    ///     AnnotatedFeature(feature:
```

```

TemporalFileSegment("file1.wav", 5 ..< 10), annotation:
"class2"),
    /// AnnotatedFeature(feature:
TemporalFileSegment("file2.wav", 2 ...), annotation:
"class1"),
    /// AnnotatedFeature(feature:
TemporalFileSegment("file3.wav", 0 ...), annotation:
"class2"),
    /// AnnotatedFeature(feature:
TemporalFileSegment("file4.wav", 0 ..< 5), annotation:
"class1")
    /// ]
    ///
    ///
    /// - Parameters:
    /// - parameters: Annotation parameters for using specific column
names from the data frame.
    /// By default the data frame is expected to have columns `filePath` and `annotation`.
    /// - continueOnFailure: A Boolean value indicating whether to
continue reading the dataframe
    /// after encountering a row that is invalid. The default value is `false`.
    /// - Returns: A list of annotated features containing
`URLRange<Double>` as feature and `Annotation` as annotation.
public func loadRangedAnnotations Annotation
DataFrameTemporalAnnotationParameters Annotation
    Bool false throws
        AnnotatedFeature TemporalFileSegment Annotation where
        Annotation Equatable Annotation Sendable

```

extension Sequence

```

    /// Returns an array containing the results of mapping the given closure over
the sequence's features.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func mapFeatures Input Output
Annotation _ Input throws Output rethrows
    AnnotatedFeature Output Annotation where Self Element
    AnnotatedFeature Input Annotation

    /// Returns an array containing the results of mapping the given async
closure over the sequence's features.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

@inlinable public func mapFeatures Input Output
Annotation _ Input async throws Output async
rethrows AnnotatedFeature Output Annotation where
Self Element AnnotatedFeature Input Annotation

```

```
    /// Returns an array containing the results of mapping the given closure over  
the sequence's annotations.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        @inlinable public func mapAnnotations Feature Input  
Output _ Input throws Output rethrows  
AnnotatedFeature Feature Output where Self Element  
AnnotatedFeature Feature Input
```

```
    /// Returns an array containing the results of mapping the given async  
closure over the sequence's annotations.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        @inlinable public func mapAnnotations Feature Input  
Output _ Input async throws Output async  
rethrows AnnotatedFeature Feature Output where  
Self Element AnnotatedFeature Feature Input
```

extension LazySequence

```
    /// Returns a lazy sequence where the elements of the result are computed  
each time they
```

```
    /// are read by calling transform function on the feature of an annotated  
feature.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        @inlinable public func mapFeatures Input Output  
Annotation _ @escaping Input Output  
LazyMapSequence Base AnnotatedFeature Output Annotation  
where Base Element AnnotatedFeature Input Annotation
```

```
    /// Returns a lazy sequence where the elements of the result are computed  
each time they
```

```
    /// are read by calling transform function on the annotation of an annotated  
feature.
```

```
    @available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
        @inlinable public func mapAnnotations Feature Input  
Output _ @escaping Input Output  
LazyMapSequence Base AnnotatedFeature Feature Output where  
Base Element AnnotatedFeature Feature Input
```

extension Sequence

```
    /// Generates two AnnotatedFeatures by randomly splitting the elements of  
the sequence, at the same proportion within each unique Annotation.
```

```
    /// - Parameters:
```

```
/// - proportion: A proportion in the range `[0.0, 1.0]`.
/// - seed: A seed number for a random-number generator.
/// - Returns: A tuple of arrays.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func randomSplit Feature Annotation
    Double           Int      nil
    AnnotatedFeature Feature Annotation
    AnnotatedFeature Feature Annotation      where Annotation
    Hashable   Self Element  AnnotatedFeature Feature
    Annotation

    /// Generates two generic arrays by randomly splitting the elements of the
    sequence.
    /// - Parameters:
    ///   - proportion: A proportion in the range `[0.0, 1.0]`.
    ///   - seed: A seed number for a random-number generator.
    /// - Returns: A tuple of array slices.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func randomSplit T           Double
Int      nil      ArraySlice T  ArraySlice T  where T
Self Element

    /// Generates two AnnotatedFeatures by randomly splitting the elements of
    the sequence, at the same proportion within each unique Annotation.
    /// - Parameters:
    ///   - proportion: A proportion in the range `[0.0, 1.0]`.
    ///   - generator: A random-number generator.
    /// - Returns: A tuple of arrays.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func randomSplit Feature Annotation Generator
    Double           inout Generator
    AnnotatedFeature Feature Annotation
    AnnotatedFeature Feature Annotation      where Annotation
    Hashable   Generator RandomNumberGenerator Self Element
    AnnotatedFeature Feature Annotation

    /// Generates two generic arrays by randomly splitting the elements of the
    sequence.
    /// - Parameters:
    ///   - proportion: A proportion in the range `[0.0, 1.0]`.
    ///   - generator: A random-number generator.
    /// - Returns: A tuple of array slices.
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0

public func randomSplit T Generator
Double           inout Generator      ArraySlice T
ArraySlice T  where T      Self Element Generator
```

RandomNumberGenerator

extension DataFrame

```
/// Creates a data frame from a sequence of annotated features.  
///  
/// - Parameters  
///   - featuresColumnID: The features column ID in the data frame.  
///   - annotationsColumnID: The annotations column ID in the data frame.  
@available macOS 13.0 iOS 16.0 tvOS 16.0 watchOS 11.0
```

```
public init S Feature Annotation _ S  
    ColumnID Feature  
ColumnID Annotation throws where S Sequence S Element  
AnnotatedFeature Feature Annotation
```