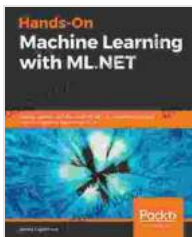


Hands-On Machine Learning with ML.NET: A Comprehensive Guide

Machine learning (ML) is a powerful technique that allows computers to learn from data and make predictions. ML.NET is a Microsoft framework that makes it easy to build and deploy ML models in .NET applications.



Hands-On Machine Learning with ML.NET: Getting started with Microsoft ML.NET to implement popular machine learning algorithms in C# by Jarred Capellman

★★★★☆ 4.1 out of 5

Language : English
File size : 9428 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 296 pages



This guide will provide you with a comprehensive overview of hands-on ML with ML.NET. We will cover topics such as:

* Data preparation * Model training * Model deployment

By the end of this guide, you will have a solid understanding of how to use ML.NET to build and deploy your own ML models.

Data Preparation

The first step in any ML project is to prepare your data. This involves cleaning the data, removing outliers, and normalizing the data.

ML.NET provides a number of tools to help you with data preparation. The following table lists some of the most common data preparation tasks and the ML.NET tools that you can use to perform them:

Task	ML.NET Tool
Data cleaning	<code>DataCleaner</code>
Outlier removal	<code>OutlierDetector</code>
Data normalization	<code>Normalizer</code>

For more information on data preparation with ML.NET, please refer to the following documentation:

* [Data Preparation with ML.NET](<https://docs.microsoft.com/en-us/dotnet/machine-learning/data-preparation/>)

Model Training

Once your data is prepared, you can begin training your ML model. ML.NET supports a variety of ML algorithms, including:

- * Linear regression
- * Logistic regression
- * Decision trees
- * Random forests
- * Support vector machines
- * Neural networks

The algorithm that you choose will depend on the type of data that you have and the task that you are trying to solve.

To train a model with ML.NET, you will first need to create a `LearningPipeline`. The learning pipeline is a sequence of steps that are used to train the model. The following code shows an example of a learning pipeline for a linear regression model:

```
csharp var learningPipeline = new LearningPipeline { Transforms = new
IEstimator[] { new DataNormalizationEstimator(normalize: true),new
ColumnConcatenatorEstimator(new[] { "Feature1", "Feature2" }) }, Trainer =
new LinearRegressionTrainer() };
```

Once you have created a learning pipeline, you can train the model by calling the `Fit` method. The following code shows an example of how to train a model using a `LearningPipeline`:

```
csharp var model = learningPipeline.Fit(data);
```

For more information on model training with ML.NET, please refer to the following documentation:

* [Model Training with ML.NET](<https://docs.microsoft.com/en-us/dotnet/machine-learning/model-training/>)

Model Deployment

Once your model is trained, you can deploy it to a production environment. ML.NET provides a number of tools to help you with model deployment, including:

* The Model Builder tool * The Model Management Service * The Azure Machine Learning service

The Model Builder tool is a graphical user interface (GUI) that makes it easy to deploy models to a variety of platforms, including:

* Azure Container Registry * Azure Kubernetes Service * Azure Web Services

The Model Management Service is a cloud-based service that provides a centralized repository for managing and deploying models.

The Azure Machine Learning service is a cloud-based service that provides a comprehensive set of tools for developing and deploying ML models.

For more information on model deployment with ML.NET, please refer to the following documentation:

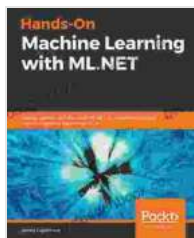
* [Model Deployment with ML.NET](https://docs.microsoft.com/en-us/dotnet/machine-learning/model-deployment/)

This guide has provided you with a comprehensive overview of hands-on ML with ML.NET. We have covered topics such as data preparation, model training, and model deployment.

By following the steps outlined in this guide, you can build and deploy your own ML models using ML.NET.

For more information on ML.NET, please visit the following website:

* [ML.NET](https://dotnet.microsoft.com/ml-net/)



Hands-On Machine Learning with ML.NET: Getting started with Microsoft ML.NET to implement popular machine learning algorithms in C# by Jarred Capellman

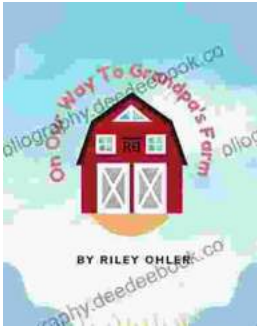
★★★★☆ 4.1 out of 5

Language : English
File size : 9428 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled

Print length : 296 pages

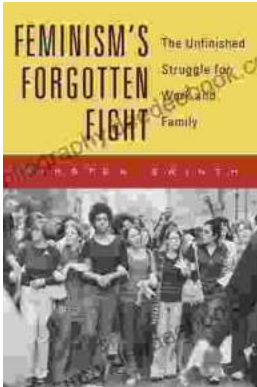
FREE

DOWNLOAD E-BOOK



Off to Grandpa's Farm: A Whimsical Adventure into the Heart of Family, Farm Life, and Nature's Embrace

Off to Grandpa's Farm is a delightful and heartwarming children's book that captures the essence of family, farm...



Feminism's Forgotten Fight: The Ongoing Battle for Economic Equality

The feminist movement has historically fought for a wide range of issues, including the right to vote, access to education, and reproductive rights. However, one of the most...