



MACHINE LEARNING WITH .NET

Asmaliza Ahzan @ Emma
asmaliza@iverson.com.my

USING ML.NET WITH ASP.NET CORE

CHAPTER 9

UNDERSTANDING THE ASP.NET CORE ARCHITECTURE

- At a high level, ASP.NET Core builds on top of .NET Core, providing a fully-featured web framework.
- As with .NET Core, ASP.NET Core runs on Windows, Linux, and macOS, in addition to allowing deployments to x86, x64, and Advanced RISC Machine (ARM) CPU architectures.
- A typical ASP.NET Core application includes the following:
 - Models
 - Views
 - Controllers

CONTROLLERS

- Controllers provide the server-side code for handling business logic for both web applications and REST services.
- Controllers can include both web and REST calls in the same controller. However, common practice is to put them in separate classes.

MODELS

- Models provide the container of data from the Controller to the View, and vice versa.
- For example, take a listing page pulling data from a database.
- The controller would return a model populated with that data, and if that same data was then used for filtering, it would also be serialized into JavaScript Object Notation (JSON) and sent back to the Controller.

VIEWS

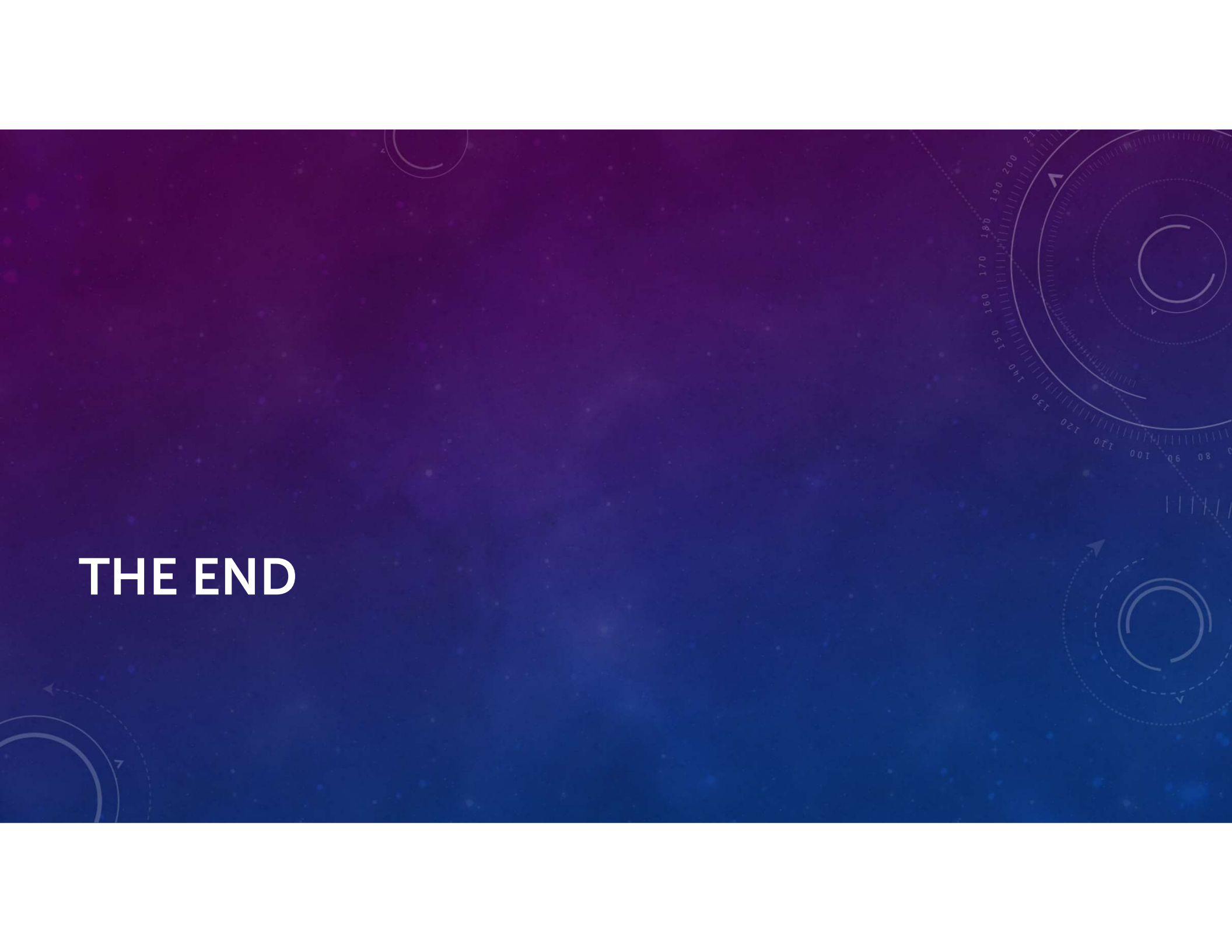
- Views provide the templates for the frontend view with support for model binding.
- Model binding allows properties bound to various Domain Object Model (DOM) objects – such as textboxes, checkboxes, and dropdowns—to be cleanly mapped to and from.
- This approach of model binding has the added benefit of supporting strongly typed references, which comes in extremely handy when you have a complex View with dozens of properties bound to a Model.

BLAZOR

- Building on the ASP.NET Core infrastructure, Blazor focuses on removing one of the biggest hurdles with complex web applications - JavaScript.
- Blazor allows you to write C# code instead of JavaScript code to handle client-side tasks such as form handling, HTTP calls, and asynchronously loading data.
- Under the hood, Blazor uses WebAssembly (Wasm), a popular high-performant JavaScript framework supported by all current browsers (Edge, Safari, Chrome, and Firefox).

CREATING THE FILE CLASSIFICATION WEB APPLICATION

- The application we will be creating is a file classification web application.
- We will be integrating machine learning with ML.NET into the web application, where an end user can upload files for classification, returning either clean or malicious files, along with a confidence of that prediction.



THE END