Application code explanation

# HARApp project folder explanation

When you in the HARApp folder, you’ll see 3 other folders and a sln file.

The **HARApp.sln** is the project file. By double clicking it, you open the project in Visual Studio.

The **HARApp** folder is the folder with de global content in. So, this is completely cross-platform.

The **HARApp.Android** is the folder specific for all Android related code.

The **HARApp.iOS** is the folder specific for all iOS related code. I haven’t done anything inside this folder.

# HARApp subfolder

In the folder **“Assets”** is the **“UCIDataset.csv”** file. This file is used in the **“UCIData.cs”** located in the **“Model”** folder. In this file is the data put into a Class object and converted to the write data frames. These dataframe can later be used as input for the tflite model.

**“MainPage.xaml”** describes the visual appearance of the main page, the page you see first the application opens. The **“MainPage.xaml.cs”** is the C# code behind the **“MainPage.xaml”** to make the page do somethings.

This is the same for the **“ResultPage.xaml”** and **“ResultPage.xaml.cs”** files in the **“View”** folder. When you click on one of the buttons on the **“MainPage.xaml”** , you get redirected to the **“ResultPage.xaml”**.

# HARApp.Android

In the folder **“Assets”** is the **“LSTM\_V6\_model4.tflite”** file. This get used in the **“TensorflowClassifier.cs”** file.

There are 3 function in that file. The function you should call from outside is the **“Classify”** function. In this function you first load the model with the **“GetModelAsMappedByteBuffer”** function. Then the model gets transformed in a usable form. After that the **“GetDataframeAsByteBuffer”** should be called. And as last step, the model should be fit with the loaded dataframe. The result in caught in the **“classificationResult”** variable.

The result should then be sent back to the **“ResultPage.xaml”** and display the result.