

PROJECT SPECIFICATION

Capstone - Azure Machine Learning Engineer

Project Setup and Style

CRITERIA	MEETS SPECIFICATIONS
The project demonstrates the use of an external dataset	<p>The project should make use of a dataset that is not available in the azureml framework. The selected dataset should be uploaded and accessed inside the AzureML Studio. The dataset should be then used to train the models used by the student.</p> <p>Note: The exact method of uploading and accessing the data is not important.</p>

The project contains detailed documentation

A README file is included in the project root and has:

- An overview of the project
- An overview of the dataset used
- An overview of the method used to get the data into your Azure ML Studio workspace.
- An overview of your AutoML experiment settings and configuration
- An overview of the types of parameters and their ranges used for the hyperparameter search
- An overview of the two models with the best parameters
- An overview of the deployed model and instructions on how to query the endpoint with a sample input
- A short overview of how to improve the project in the future
- ALL the screenshots required with a short description
- A link to the screencast video on YouTube (or a similar alternative streaming service)

You are encouraged to make your README document as informative and detailed as possible. To do this, you can use screenshots, block diagrams and charts.

<p>The project contains a screencast video demoing the working project</p>	<p>The screencast should meet the following criteria:</p> <ul style="list-style-type: none"> • Screencast is 1-5 minutes in length • Audio is clear and understandable • Video is 1080P or higher with 16:9 aspect ratio • text is readable <p>The screencast should demonstrate:</p> <ul style="list-style-type: none"> • A working model • Demo of the deployed model • Demo of a sample request sent to the endpoint and its response • Demo of any additional feature of your model
--	---

AutoML Model

CRITERIA	MEETS SPECIFICATIONS
<p>The code demonstrates training of model using AutoML</p>	<p>The project should include:</p> <ul style="list-style-type: none"> • AutoML settings (the settings you use are up to you) • An AutoML config (The experiment configuration is up to you) <p>You are encouraged to provide details behind your reasoning for choosing the settings and experiment configuration</p>

The code visualizes the progress of the hyperparameter tuning runs	<p>The submission contains a screenshot of the RunDetails widget that shows the progress of the training runs of the different experiments.</p> <p>You are encouraged to provide details in the submitted Jupyter notebook about the performance of the different models on the primary metric of your experiment.</p>
The code displays the properties of the best trained model	Your submitted Jupyter notebook contains details of the best model and documents the parameters of that model.
The code demonstrates the saving of the best model from all the experiments	The submitted notebook contains code showing the best model being registered and includes a screenshot of the best model with its run id.

Hyperdrive Model

CRITERIA	MEETS SPECIFICATIONS
The code demonstrates the use of hyperdrive to tune the hyperparameters of a model	<p>The project should use the following features of hyperdrive:</p> <ul style="list-style-type: none"> • Use one type of sampling method: grid sampling, random sampling or Bayesian sampling. • Logs metrics during the training process • Specify an early termination policy (not required in case of Bayesian sampling).

<p>The code demonstrates tuning of at least 2 hyperparameters using hyperdrive</p>	<p>The project should defines a search space for at least 2 different hyperparameters to be tuned.</p> <p>The specific model is not important as long as a search is done over at least two different hyperparameters of the model. However, do not limit yourself to just two parameters, you are encouraged to</p> <p>You are encouraged to provide documentation in the submitted Jupyter notebook of the different hyperparameters that were optimized and why they were chosen.</p>
<p>The code visualizes the progress of the hyperparameter tuning runs</p>	<p>The submission contains a screenshot of the RunDetails widget that shows the progress of the training runs of the different experiments.</p> <p>You are encouraged to provide details in the submitted Jupyter notebook about the effects of the different hyperparameters on the primary metric of your model.</p>
<p>The code demonstrates the saving of the best model from all the experiments</p>	<p>The submission includes a screenshot of the best model with its run id and the different hyperparameters that were tuned. The submitted notebook also contains code showing the best model being registered.</p>

Deploying the Model

CRITERIA	MEETS SPECIFICATIONS
The project demonstrates the deployed model	<p>Your project should contain the following:</p> <ul style="list-style-type: none"> • Model being registered • Model being deployed • A file containing the environment details <p>The submission contains a screenshot showing the model endpoint as active.</p>
The code demonstrates a request being sent to the active endpoint	Your submission should contain code showing an inference request being sent to the deployed model.

Suggestions to Make Your Project Stand Out!

1. Convert your model to ONNX format. Your Jupyter notebook contains documentation and code for converting the model to the ONNX format.
2. Deploy your model to the Edge using Azure IoT Edge. Your submission includes an extra Jupyter notebook containing the code for deploying the model to the edge. The README file contains documentation about the deployment as well as screenshots showing the data arriving at your IoT hub. You can use [this tutorial](#) for reference.
3. Enable logging in your deployed web app. Can you log useful data about the requests being sent to the webapp? This may include, inference time, the time at which the request arrived, and other details. You can find out more about logging [here](#). Your submission should contain code to perform logging in the deployed web app. Your **README** file also contains information about

the details logged and a screenshot of the logs and metrics collected.