

Building a Model to Predict Text Sentiment on Microsoft Azure

There are many ways to build a model that predict natural language text sentiment on Azure. After using Azure quite intensively these past few days, I would say I am impressed about what the platform offers for Machine Learning and Analytics Use Cases.

The objective of our Use Case is to predict tweets Sentiment using the **sentiment140** dataset. It contains 1 600 000 tweets extracted using the twitter api . The tweets have been annotated (0 = negative, 1 = positive) - *a customized version of the original data*, and they can be used to detect sentiment. Through the project we tested many ways of attaining this objectif through the use of Azure Services. We encountered three ways: the use of the Text Analytics API from Azure Cognitive Services; the use of Azure Machine Learning Studio; and the creation of a Custom Model to train and deploy on Azure. But what are the pros and cons of those ways of using Azure Machine Learning services for our Use Case? Is one method better than the two others? It is tricky to answer. Let's dive deeper by explaining what we learned from using those services.

Deep learning algorithms are very computer intensive and require large enough datasets to get good accuracies. Therefore, if you are in the case of small data and resource constraints, you are better off using an API. An API just is a programmed service that comes with its methods and tools to ease the use of it. It is very convenient for application integration. For instance with the Text Analytics API we used the Azure ML SDK to integrate it with our local jupyter Notebook and interact with Azure Cognitive Services. APIs are very practical for programmers. Using the Text Analytics Service helped us to build a quick Proof Of Concept to decide on the usefulness of a sentiment détection problem.

In our Use Case we have enough data, but if we need to see if building a text sentiment application worth it, trying the Text Analytics service could be definitely a good move. Doing that, we need to be aware of the fact that we do not have access to more information on the Text Analytics service. It could be gender or race biased. With that said, if we need to go further and try something more adapted to our use case, to our data, but not go deeper, since we may not have enough skills in Machine Learning or Data Science in general, we could try Azure Machine Learning Studio.

Azure ML Studio is a very simple and handy machine learning software application that helps to build an end-to-end ML Use Case by dragging and dropping boxes. And that from uploading your data in the software to deploying a web service that offers you the possibility to have a full fledged application service running and serving your data in real time or in batch mode. The limitation of this service is that even though it offers the possibility to insert your own Python or R code in the pipeline, you cannot use large dataset and do not have a hand on the choice of the resource to be used for your compute (this is bypassed by the Azure Machine Learning). Therefore, to have full control of your model and enjoy compute resources flexibility you need to build a custom model, your own Machine Learning Model, and then use Azure ML SDK tools kits, a bunch of libraries easy to install and implement from your local machine.

As a programmer, you want to have full control in the building of your machine learning model. Azure ML SDK is the solution. It offers you the possibility to build and test (even

deploy) your model using its library in your local machine, and then when you are ready, you have full confidence in your model, you can contact Azure Services, through the SDK (or graphically by using the Azure Portal), to train your model in a larger scale and deploy it in production.

To summarize, there is no better method, it depends not only on your Use Case, but the amount of data you have for building your model, and your Data Science skills.