

MICROSOFT-ZERTIFIZIERUNG

Datenwissenschaftler werden

Ein .NET-Entwickler bildet sich fort und staunt.

Früher war die Welt für .NET-Entwickler übersichtlich. Man lernte C# (oder Visual Basic) und nahm sich die Windows- und Web-Komponenten des .NET Framework vor. Vielleicht setzte man sich auch mit SQL Server auseinander. Dazu bot Microsoft passende Zertifizierungen an, nämlich den Microsoft Certified Technology Specialist (MCTS) für Windows-, Web- und Datenbankentwicklung. Eventuell noch die eine oder andere Spezialität, Business Intelligence zum Beispiel. Damit ließ sich nach meinem damaligen Empfinden so ziemlich alles an Softwarelösungen herstellen, was das Herz beehrte. Der Tellerrand war klar definiert. Warum drübergucken? Schöne .NET-Welt.

Nur hat die Welt die Angewohnheit, sich zu verändern, und während in den letzten Jahren viele Technologien gekommen und gegangen sind, scheint der aktuelle Trend zu Data Science und künstlicher Intelligenz wohl zu bleiben. Es handelt sich dabei ja auch um eine völlig neue Art, Probleme zu lösen: Anwendungen (Modelle) werden nicht programmiert, sondern durch Training angelernt. Man schreibt zwar (meistens) noch Code, muss aber völlig umdenken. Es geht hier nicht um die Entwicklung exakter Algorithmen, sondern um die Verarbeitung von Daten aus einer unordentlichen Welt.

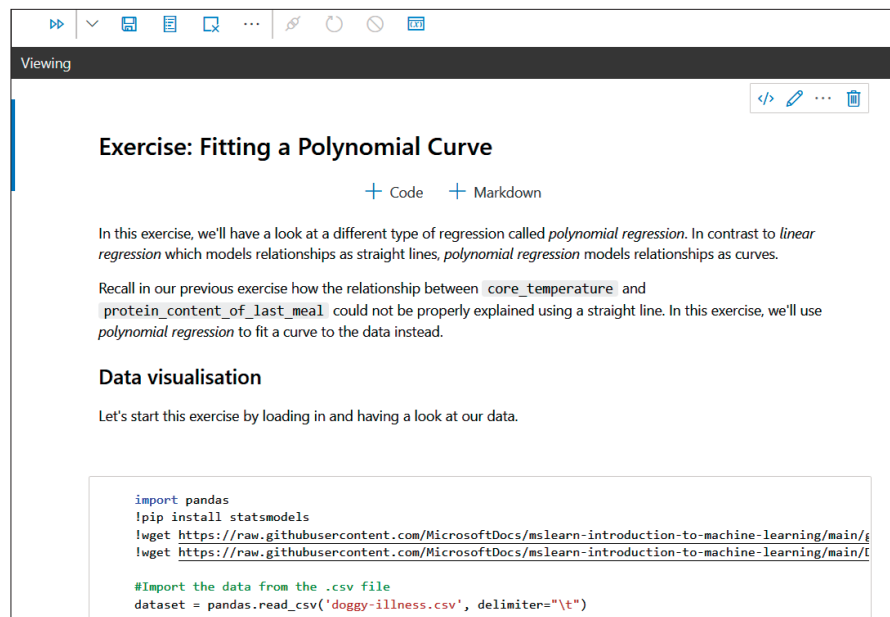
Dem Trend folgend hat Microsoft eine Vielfalt an Data-Science-Technologien herausgebracht, darunter ML.NET, Windows ML, SQL Server Machine Learning Services, eine virtuelle Maschine für Data Science, eine Machine-Learning-Erweiterung für Azure Data Studio, Azure Machine Learning Services, Azure Cognitive Services, Azure Synapse Analytics und Azure Databricks [1]. Außerdem gibt es den ML.NET-Modell-Generator für Visual Studio [2].

Die Zertifizierung

Nun gut, ich habe die Zeichen der Zeit erkannt: Ich will mich fortbilden und meine Software intelligent machen. Vielleicht soll ja im nächsten Projekt ein smarterer Karl Klammer [3]

entwickelt werden. Die vermeintlich passende Microsoft-Zertifizierung ist auch schnell gefunden: Azure Data Scientist Associate [4]. Und weil „Datenanalyse“ auch nicht schlecht klingt, nehme ich noch den Power BI Data Analyst Associate [5] dazu.

Für jede Prüfung stellt Microsoft eine Liste der Prüfungsinhalte zur Verfügung, sodass ich weiß, auf welche Themen ich

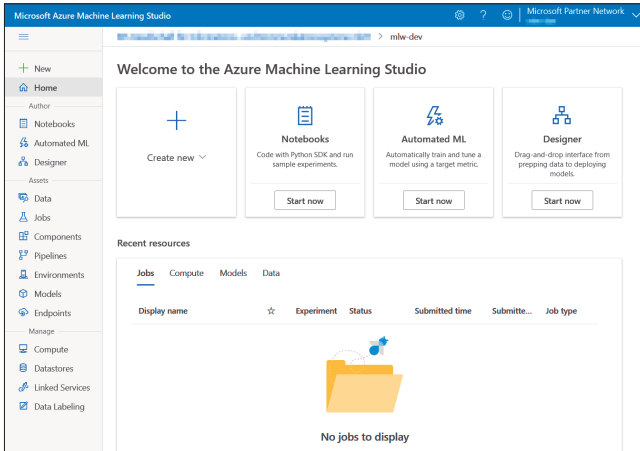


Ausschnitt aus einem Übungs-Notebook mit Markdown-Text und Python-Code (Bild 1)

mich vorbereiten muss. Dazu kann ich die angebotene Schulungsdokumentation nutzen; erfahrungsgemäß reicht sie allerdings nicht aus. Detailfragen lassen sich nur mithilfe der offiziellen technischen Dokumentation beantworten.

Man findet im Grunde alles Nötige, wenn man sich in der technischen Microsoft-Dokumentation [6] zu den passenden Themen durchhangelt. Es gibt auch von Drittanbietern sehr hilfreiche und nicht allzu kostspielige Angebote für die Prüfungsvorbereitung, wie zum Beispiel auf Udemy [7] oder Coursera [8].

Inhaltlich werden bei der ersten von mir gewählten Zertifizierung, dem Data Scientist Associate, folgende Themenbereiche behandelt:



Die Startseite des Azure Machine Learning Studio gibt einen Eindruck davon, was möglich ist (Bild 2)

- Grundlagen Data Science und Maschinelles Lernen
- Visuelle Tools in Azure Machine Learning
- Erstellen und Betreiben von Machine-Learning-Lösungen mit Azure Machine Learning
- Erstellen und Betreiben von Machine-Learning-Lösungen mit Azure Databricks

Je nach Vorkenntnissen kann man auch einzelne Bereiche überspringen; dazu bietet Microsoft drei verschiedene Lernoptionen an.

Ich lege los ...

Das ist alles ganz spannend, ich beginne also mit allgemeinen Grundlagen und vielen praktischen Beispielen, die – für mich etwas überraschend – ausschließlich in Python daher kommen. Ausgeführt wird der Code in Notebooks (Bild 1). Diese sind den Jupyter Notebooks [9] nachempfunden, also interaktiven Dokumenten, die sowohl formatierten Text als auch Programmcode enthalten können. Sie werden in der Data-Science-Welt schon länger eingesetzt.

Anschließend arbeite ich im Azure Machine Learning Studio (Bild 2); hier beschäftige ich mich mit automatischem maschinellem Lernen (AutoML), visuellen Tools und wiederum Python-Notebooks. Neben Azure-eigenen Bibliotheken aus dem Namensraum *azureml* wird mit etablierten Technologien wie pandas [10], scikit-learn [11], PyTorch [12] und TensorFlow [13] gearbeitet.

Und schließlich ist Azure Databricks [14] dran. Bei Databricks handelt es sich um eine Plattform für hochskalierbare Data-Science-Anwendungen, die von Microsoft in Azure integriert wurde. Die Funktionalität ähnelt stark jener des Machine Learning Studio (Bild 3). Und programmiert wird – ja, genau – in Python.

An dieser Stelle wundere ich mich, dass das Ganze schon vorbei ist. Technisch gesehen wird tatsächlich nur Azure Machine Learning mit dem Python SDK und Databricks abgedeckt. Das für .NET-Entwickler besonders interessante ML.NET kommt in der Schulung (ebenso wenig wie in der Prüfung) nicht vor. ▶

HANDS-ON-WORKSHOPS UND WEITERBILDUNG FÜR SOFTWARE-ENTWICKLER UND -ARCHITEKTEN

TRAININGS



Clean Code und
Software Design
Trainer: David Tielke



Progressive Web
App Bootcamp
Trainer: Peter Körner



Einstieg in Python
Trainer: David Pinezich



Continuous
Delivery & Integration
Trainer: Stephan M. Rossbach



Git-Masterclass
Trainer: Michael Kaufmann



SignalR und
Event-Signaling
Trainer: Patrick Schnell

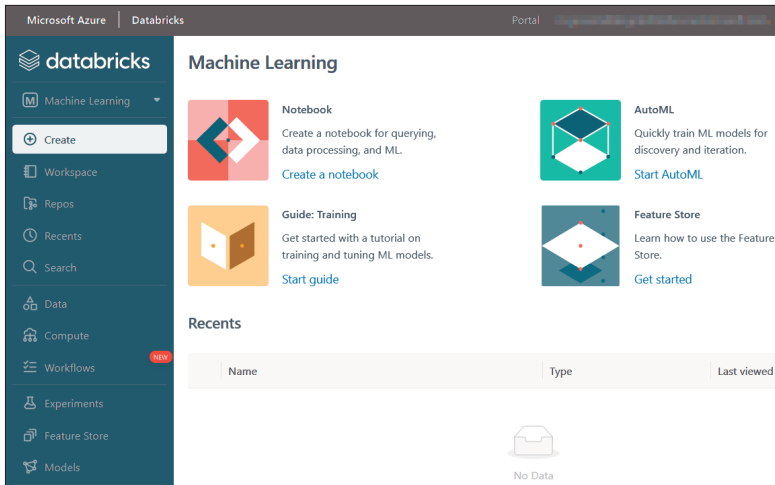


SQL-Server-
Programmierung
Trainer: Thorsten Kansy

REMOTE
TRAININGS MÖGLICH!

developer-academy.de

Ihre Ansprechpartnerin: Susanne Herl
+49 731 88005-8835 • susanne.herl@ebnermedia.de



Azure Databricks ähnelt dem Machine Learning Studio (Bild 3)

Genau genommen bleibt .NET komplett außen vor. Der ML.NET-Modell-Generator spielt keine Rolle. Und SQL Server Machine Learning bleibt unerwähnt. Dementsprechend wird auch kein C# oder SQL eingesetzt. Es wird nichts kompiliert, sondern alles in Notebooks über Skripte erledigt.

Die zweite Schulung (zum Power BI Data Analyst Associate) ändert daran auch nichts. Wie der Name schon sagt, geht es um Power BI, Microsofts Tool für die Aufbereitung und Analyse von Daten. Das Tool richtet sich eher an Power-User als an Entwickler. Als Sprachen kommen hier in erster Linie M [15] und DAX [16] zum Einsatz.

Und nun?

Ich bin irritiert. Klar, Microsoft integriert schon länger Open-Source-Technologien in seine Produkte und pusht gegenwärtig alles Richtung Cloud. Zitat aus der Marketing-Abteilung: „Produkte sind in den Hintergrund getreten. Interagierende Lösungswelten und Services sind immer wichtiger geworden.“ Aber gar kein .NET? Das fühlt sich erst mal wie eine Null-Pointer-Exception an.

Hier spiegelt sich wohl wider, dass die Arbeit im Data-Science-Umfeld fundamental anders abläuft, als wir Entwickler es gewohnt sind. Datenerkundung geht schneller in einer Skripting-Umgebung und ressourcenintensives Trainieren von Modellen profitiert in der Tat von der Cloud. Aus Sicht von Microsoft sollen wir, die klassischen Entwickler, vermutlich gar keine Anwendungen mit ML.NET erstellen, mit denen man dann Datenwissenschaft betreiben könnte.

Vielmehr setzt Microsoft sein ML.NET in eigenen Produkten ein, die der Datenwissenschaftler benutzen soll. Dafür spricht auch die Tatsache, dass das Softwarehaus neue Funktionen zuerst in seine Tools einbaut und erst später für Entwickler freigibt. Das war zuletzt bei den ML.NET-AutoML-Funktionen so [17].

Fazit

Meiner Fortbildungsausflug brachte eine – zumindest für mich – überraschende Erkenntnis. Wer Data Science verstehen will, muss sich sogar bei Microsoft zuerst mit den etab-

lierten Open-Source-Technologien auseinandersetzen. Das betrifft insbesondere Python und einschlägige Data-Science-Bibliotheken sowie die Arbeit mit (Jupyter-)Notebooks.

Anschließend steht eine Reihe an Microsoft-Tools bereit, die sich allerdings teilweise funktionell überlappen. Bei der Zertifizierung spielen zudem nur die Azure-basierten Tools eine Rolle. Demzufolge ist die Zertifizierung zum Data Scientist Associate nur dann sinnvoll, wenn man auch im Azure-Umfeld arbeiten will.

ML.NET – als Bibliothek für Entwickler – scheint eher ein Nebenprodukt zu sein, das nicht sonderlich gefördert wird. Es kann nicht schaden, das im Hinterkopf zu behalten, wenn man ML.NET für eigene Anwendungen einsetzen möchte. ■

[1] Vergleich der Machine-Learning-Produkte und -Technologien von Microsoft, www.dotnetpro.de/SL2212Zertifizierung1

[2] ML.NET-Modell-Generator, www.dotnetpro.de/SL2212Zertifizierung2

[3] Karl Klammer (Microsoft), www.dotnetpro.de/SL2212Zertifizierung3

[4] Microsoft Certified: Azure Data Scientist Associate, www.dotnetpro.de/SL2212Zertifizierung4

[5] Microsoft Certified: Power BI Data Analyst Associate, www.dotnetpro.de/SL2212Zertifizierung5

[6] Technische Microsoft-Dokumentation, <https://docs.microsoft.com/de-de>

[7] Udemy, <https://www.udemy.com/de>

[8] Coursera, <https://www.coursera.org>

[9] Jupyter, <https://jupyter.org>

[10] pandas, <https://pandas.pydata.org>

[11] scikit-learn, <https://scikit-learn.org>

[12] PyTorch, <https://pytorch.org>

[13] TensorFlow, <https://www.tensorflow.org>

[14] Databricks, <https://databricks.com/de>

[15] Power Query M – Formelsprache, <https://docs.microsoft.com/de-de/powerquery-m>

[16] DAX-Referenz (Data Analysis Expressions), <https://docs.microsoft.com/de-de/dax>

[17] What's new with ML.NET Automated ML (AutoML) and tooling, www.dotnetpro.de/SL2212Zertifizierung6



Martin Gossen

ist IT-Berater bei der IKS GmbH in Hilden. Er erstellt seit 15 Jahren Softwarelösungen auf Basis von C#, .NET und Microsoft SQL Server. Sie erreichen ihn unter m.gossen@iks-gmbh.com.