DATA SATURDAYS #64 PARMA 2024

Unlocking Advanced Analytics in Power BI Without Premium Capacity

Luca Zavarella

LABS

Sponsor & Org

iSolutions

engage
LABS

DATA SKILLS
UNDERSTANDING THE WORLD

LIME
ware

daniela
malvisi
COMMUNICATION

Lucient
ITALIA

Bi Factory
DATA KNOWLEDGE ADVISOR

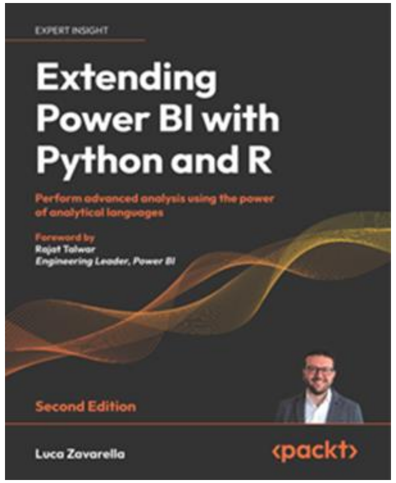
About Me

Luca Zavarella

Working in Business Intelligence with SQL Server since 2007
Microsoft MVP for Artificial Intelligence & Data Platform
Microsoft Certified: Azure Data Scientist Associate
Author of “*Extending Power BI with Python and R, 2nd Edition*” by Packt

Head of Data & AI @ icubed

Email: luca.zavarella@icubed.it
Twitter: @lucazav
LinkedIn: https://it.linkedin.com/in/lucazavarella
Blog: https://lucazavarella.medium.com


EXPERT INSIGHT
Extending Power BI with Python and R
Perform advanced analysis using the power of analytical languages
Foreword by Rajat Talwar Engineering Leader, Power BI
Second Edition
Luca Zavarella
‹packt›

Agenda

- AI in Power BI with Premium or Embedded Capacities
- Configuring SQL Server ML Services
- A closer look at `sp_execute_external_script`
- Implementing the predictive backend
- Let’s Use SQL Server External Languages with Power BI
- Let’s Put it All Into Practice in Power BI

AI in Power BI with Premium or Embedded Capacities

AI Insights Integration

Functions

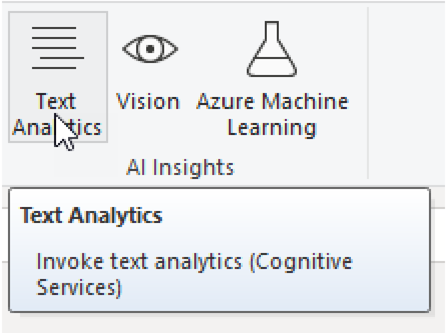
- **Language detection**: For each text field, returns the language name and ISO identifier (120 languages)
- **Extract Key Phrases**: For each text field, returns a list of key phrases
- **Score Sentiment**: Returns a sentiment score for each document, ranging from 0 (negative) to 1 (positive)
- **Tag Images**: Returns tags based on more than 2,000 recognizable objects, living beings, scenery, and actions.

Access

- Self-Service Data Prep for dataflows
- Power Query online in the Power BI service
- Functions accessed from the ribbon in Power Query Editor or by invoking the M function directly

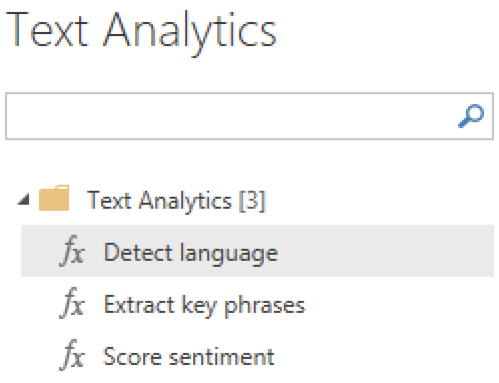
Support

- Power BI
  - EM2, A2 or P1 and above capacities
  - Embedded
- Fabric capacities: F16 and above



Text Analytics

Invoke text analytics (Cognitive Services)



Azure ML Models Integration

Functions

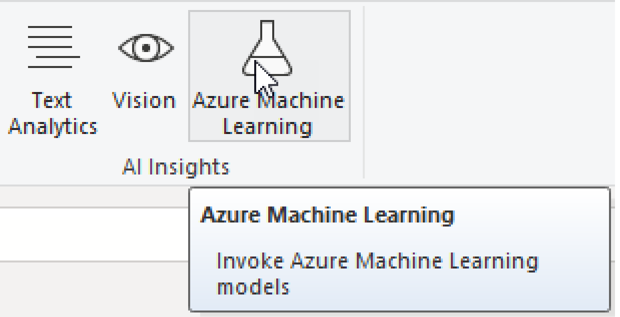
- Any type of machine learning model provided by data scientists in the Azure ML workspace
  - Deployed API shaped accordingly to be consumed by Power BI

Access

- Self-Service Data Prep for dataflows
- Power Query online in the Power BI service
- Functions accessed from the ribbon in Power Query Editor or by invoking the M function directly

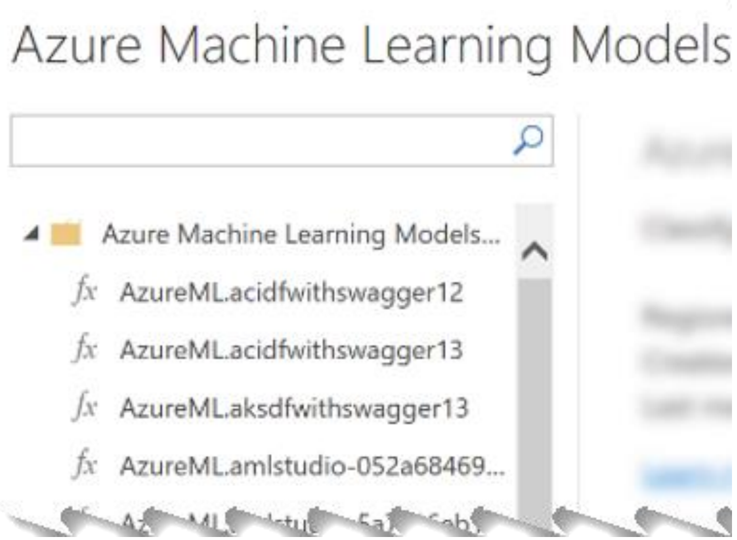
Support

- Power BI
  - EM2, A2 or P1 and above capacities
  - Embedded
- Fabric capacities: F16 and above



Azure Machine Learning

Invoke Azure Machine Learning models


Azure Machine Learning Models

Azure Machine Learning Models...
AzureML.acidfwithswagger12
AzureML.acidfwithswagger13
AzureML.aksdfwithswagger13
AzureML.amlstudio-052a68469...

AutoML Integration

Functions

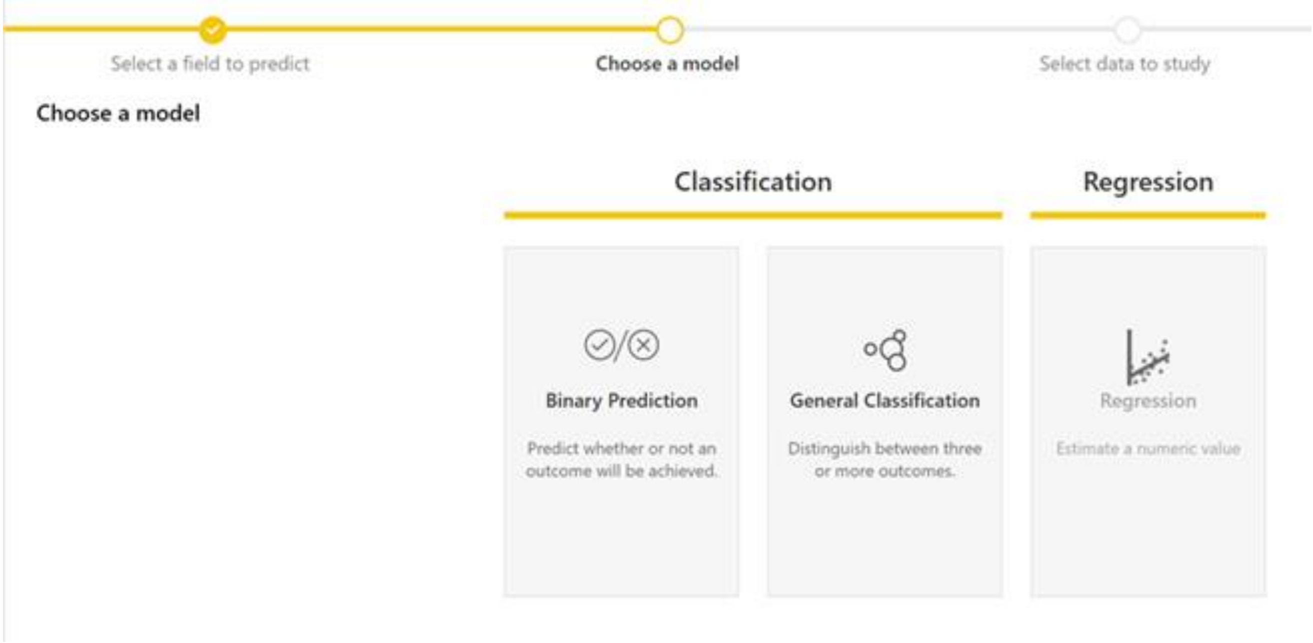
- Binary classification
- Multi-label classification
- Regression
- No time-series forecasting!

Access

- Available for Dataflows
- Premium and embedded capacities
- No need of Azure ML Workspace

Support

- Power BI
  - EM2, A2 or P1 and above capacities
  - Premium Per User (PPU) license or Embedded
- Fabric capacities: F16 and above



Choose a model

Classification

Regression

Binary Prediction

Predict whether or not an outcome will be achieved.

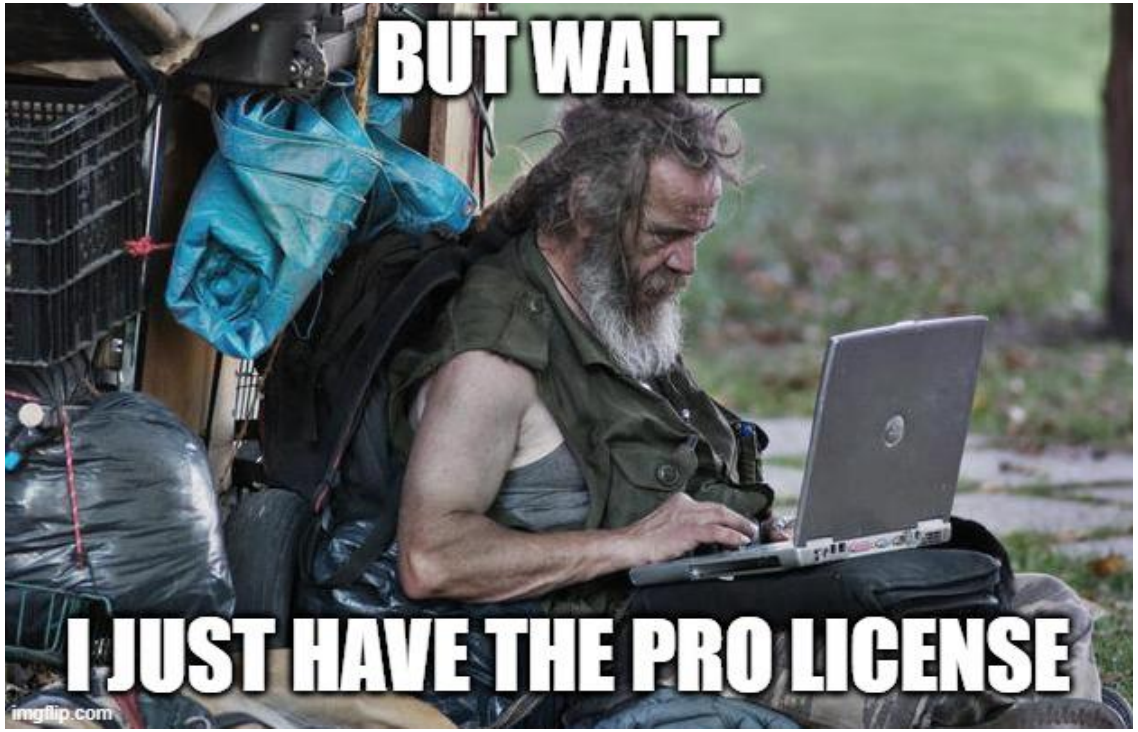
General Classification

Distinguish between three or more outcomes.

Regression

Estimate a numeric value

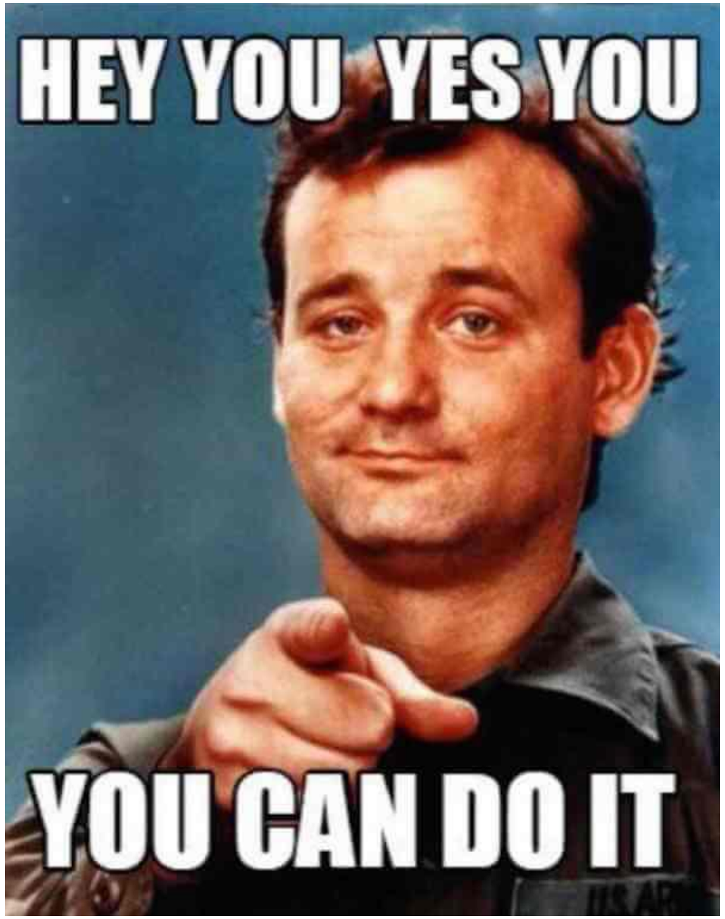
This AI Stuff is Cool, but...



Perform Advanced Analytics with a Pro License on SQL Server

Here what you need

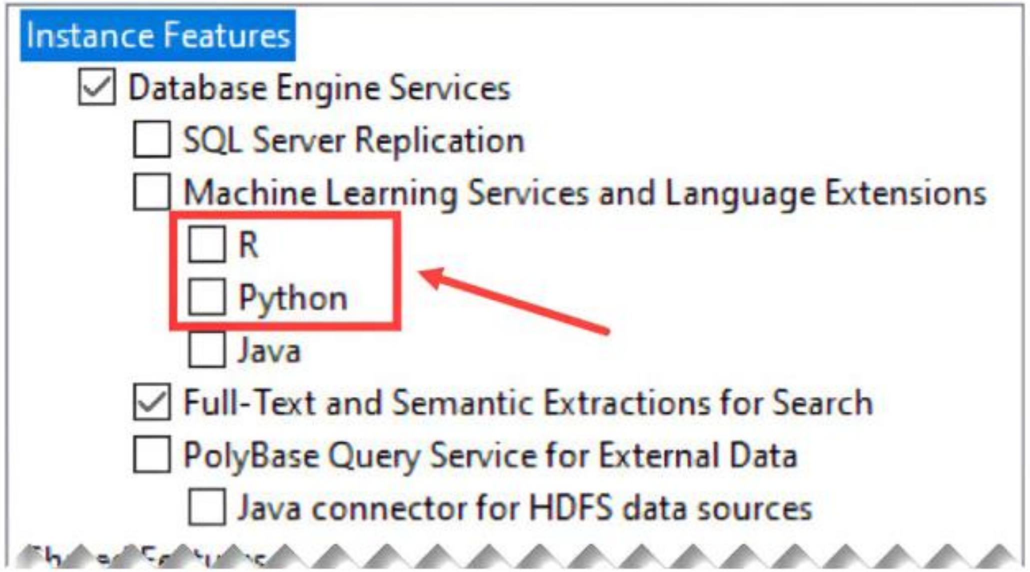
- “Tweak” SQL Server a little
- Learn a trick to call SQL Server procedures from Power BI
- A trained ML model to score predictions (for this demo)
- Understand a little of Python



Configuring SQL Server ML Services

Hidden Gems in SQL Server

- SQL Server communicates with external engines through the Launchpad service thanks to the Extensibility Framework
- R and Python installed from the Setup are proprietary components
  - Using these components the latest versions you can have in SQL Server 2019 are Python 3.7.1 and R 3.5.2
- With SQL Server 2019 CU3 the External Host process (*ExtHost.exe*) has been introduced in the Extensibility Framework
- Thanks to the External Host, you can add any external compiled language (e.g. Java, C#) using the Extensibility Framework API
- Microsoft open sourced the Language Extensions, including files also for R and Python (newer versions)

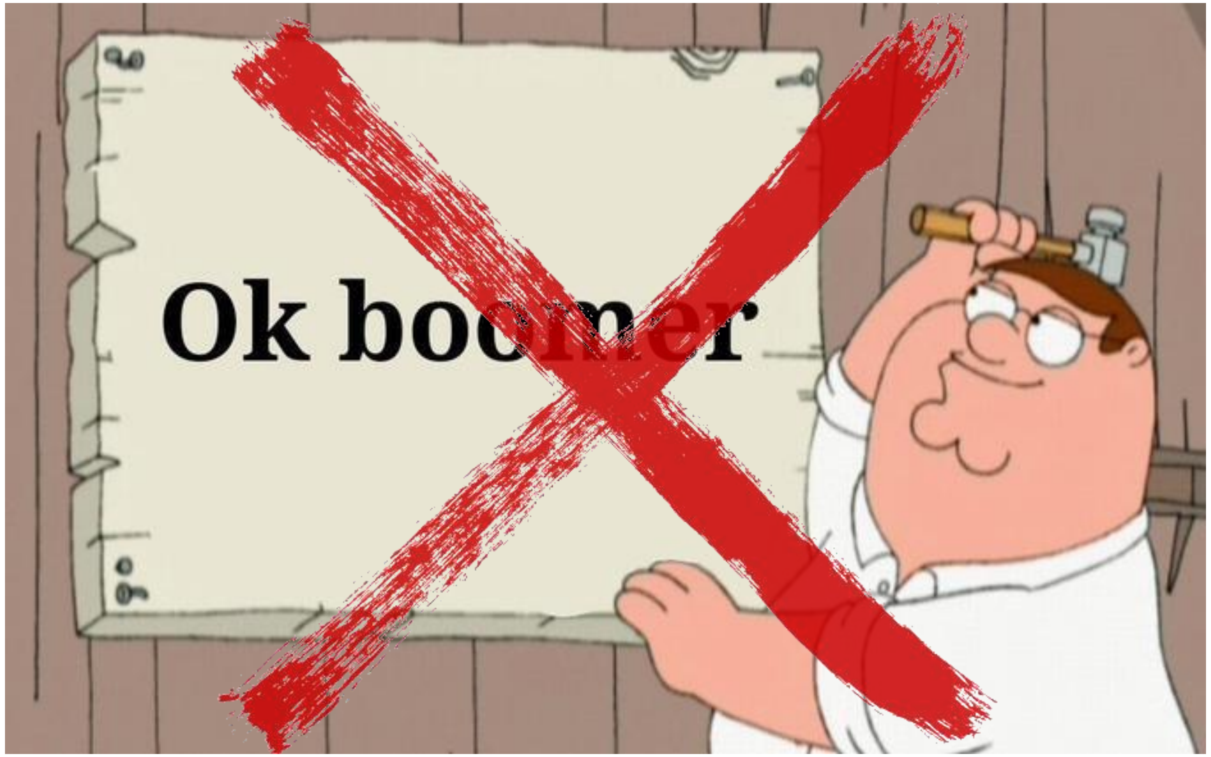


Instance Features

- ☑ Database Engine Services
  - □ SQL Server Replication
  - □ Machine Learning Services and Language Extensions
    - □ R
    - □ Python
    - □ Java
  - ☑ Full-Text and Semantic Extractions for Search
  - □ PolyBase Query Service for External Data
    - □ Java connector for HDFS data sources

Starting with SQL Server 2022, you will no longer be able to install the proprietary components for Python and R from the setup wizard; you will need to manually install the engines and manually connect them to SQL Server

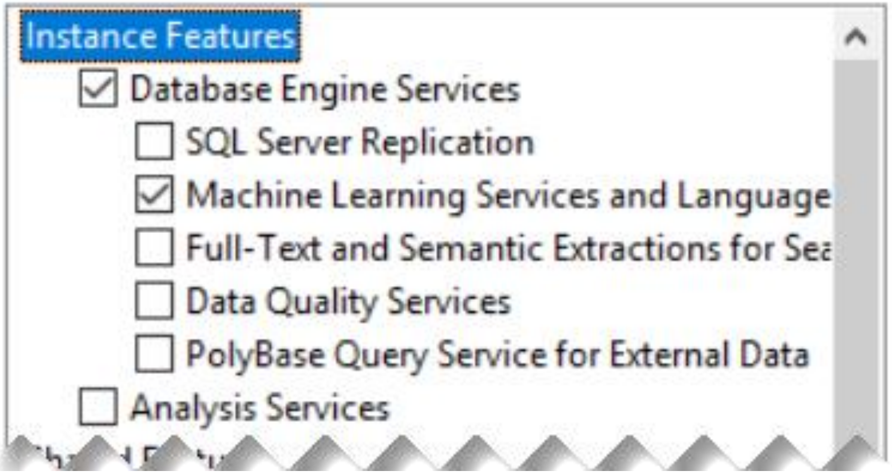
Only Available in SQL Server on-prem?



- Machine Learning Services are also available in Azure SQL Managed Instance
  - a PaaS offering with near-perfect compatibility with the latest SQL Server database engine in Enterprise Edition
- Not in ~~Azure SQL Database~~
- Not in ~~SQL Database in Fabric~~

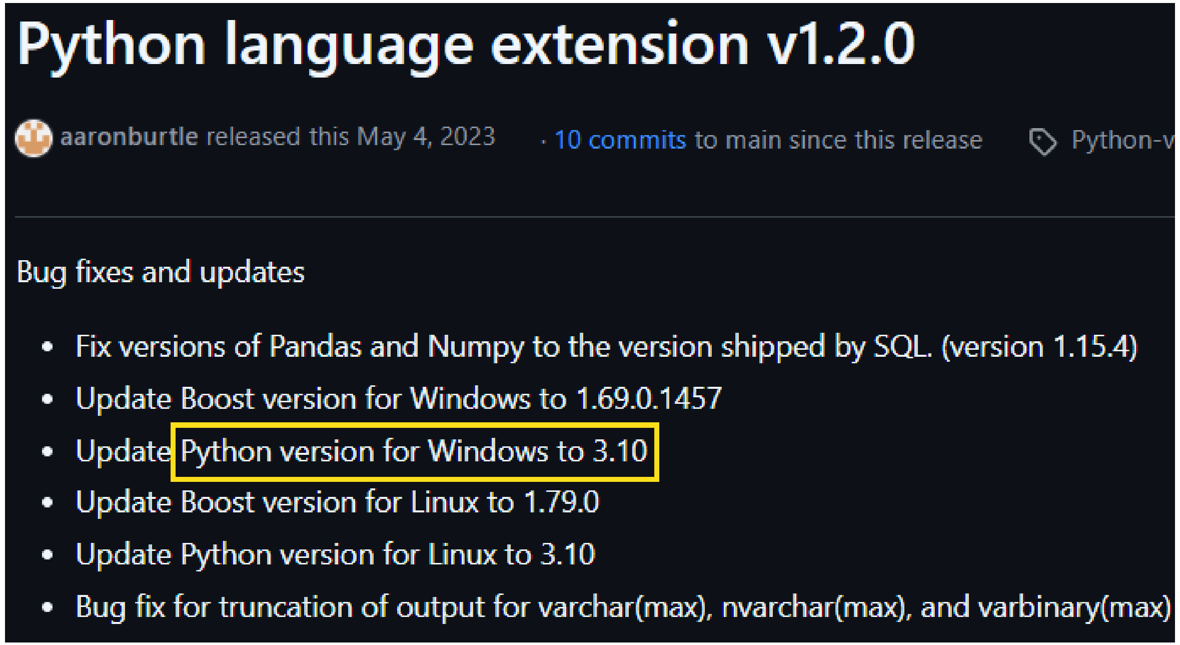
Installing Python custom runtimes for SQL Server 1/2

- You need to have at least SQL Server 2019 with at least CU3 installed
- Install only the *Machine Learning Services and Language Extensions* feature from the Setup
- To choose a Python (or R) version to install, check which one is supported by the latest Language Extension releases (https://github.com/microsoft/sql-server-language-extensions/releases)



Instance Features

- ☑ Database Engine Services
  - □ SQL Server Replication
  - ☑ Machine Learning Services and Language
  - □ Full-Text and Semantic Extractions for Sea
  - □ Data Quality Services
  - □ PolyBase Query Service for External Data
- □ Analysis Services



Python language extension v1.2.0

aaronburtle released this May 4, 2023 · 10 commits to main since this release Python-v

Bug fixes and updates

- Fix versions of Pandas and Numpy to the version shipped by SQL. (version 1.15.4)
- Update Boost version for Windows to 1.69.0.1457
- Update Python version for Windows to 3.10
- Update Boost version for Linux to 1.79.0
- Update Python version for Linux to 3.10
- Bug fix for truncation of output for varchar(max), nvarchar(max), and varbinary(max)

Installing Python custom runtimes for SQL Server 2/2

- The latest Python 3.10 release is 3.10.15 (https://www.python.org/downloads/)
- Unfortunately, only release 3.10.11 provides Windows installers (https://www.python.org/downloads/release/python-31011/)
- During the Python installation, be sure to:
  - Check “Add python.exe to PATH”
  - Check “For all users (requires admin privileges)”
  - Uncheck “Download debugging symbols”
  - Uncheck “Download debug binaries”

Configuring the SQL Server Language Extensions 1/3

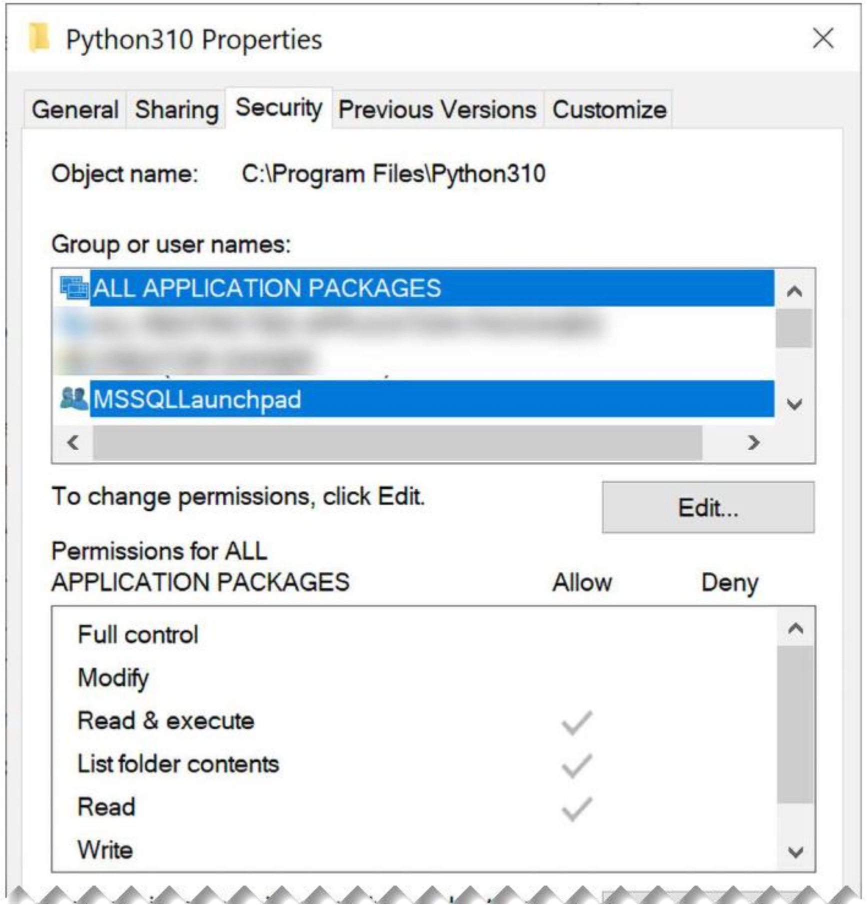
Steps to follow

1. Make the Launchpad service have read and execute (or write) permissions on the Python 3.10 installation folders and their subfolders

   -
     ```
     icacls "C:\Program Files\Python310" /grant "NT Service\MSSQLLAUNCHPAD":(OI)(CI) RX /T
     ```
2. Do the same for AppContainers, giving the same permissions to ALL APPLICATION PACKAGES (tied to AppContainers for isolation purpose)

   -
     ```
     icacls "C:\Program Files\Python310" /grant *S-1-15-2-1:(OI)(CI)RX /T
     ```

If the external language script needs to access a specific folder in the file system, you must also give the preceding permissions (for both the NT Service\MSSQLLAUNCHPAD and ALL APPLICATION PACKAGES users) to the folder that the engine needs to access.
Remember to restart the Launchpad service!



Object name: C:\Program Files\Python310

Group or user names:

- ALL APPLICATION PACKAGES
- [redacted]
- MSSQLLaunchpad

To change permissions, click Edit.

Permissions for ALL APPLICATION PACKAGES	Allow	Deny
Full control		
Modify		
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write		

Configuring the SQL Server Language Extensions 2/3

Steps to follow

1. The Language Extension file you need to download from GitHub is *python-lang-extension-windows-release.zip*
2. Make sure you save these files in a folder that SQL Server has access to (e.g. *C:\LanguageExtensions*)
3. Register the external Language Extension in your selected database from the specified file paths giving it a specific name (e.g. *py310*, case sensitive):

```
USE <your-database>
GO

CREATE EXTERNAL LANGUAGE [py310]
FROM (CONTENT = N'C:\LanguageExtensions\python-lang-extension-windows-release.zip',
    FILE_NAME = 'pythonextension.dll',
    ENVIRONMENT_VARIABLES = N'{"PYTHONHOME": "C:\\Program Files\\Python310"}');
GO
```

The CREATE EXTERNAL LANGUAGE statement creates an external language specifically in the database on which the query is executed. If you need the ability to run Python and R scripts on a different database, you must run CREATE EXTERNAL LANGUAGE queries on that database too

Configuring the SQL Server Language Extensions 3/3

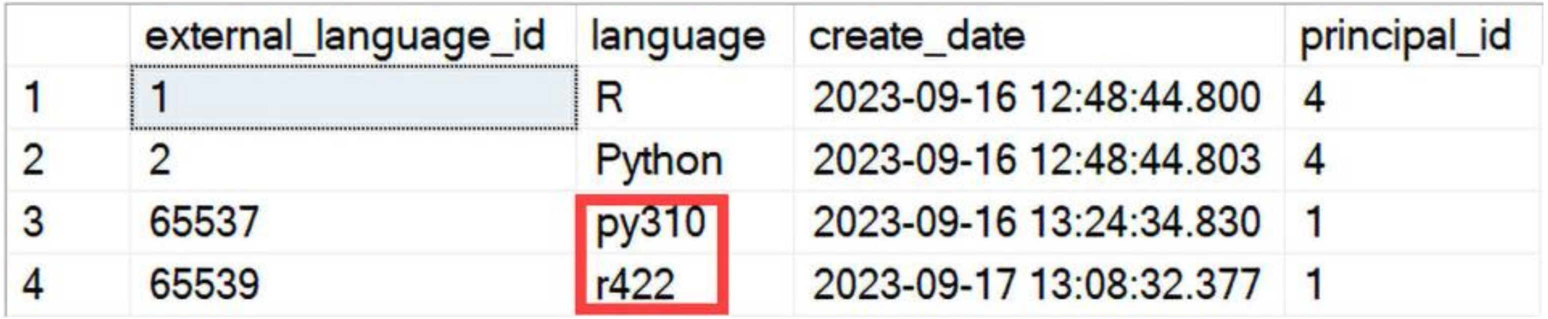
Steps to follow

1. You need to enable external scripts in your instance with this script:

```
sp_configure 'external scripts enabled', 1;
RECONFIGURE WITH OVERRIDE;
```

2. Check if the external language has been created correctly using the *sys.external_languages* catalog view:

```
SELECT * FROM sys.external_languages;
```

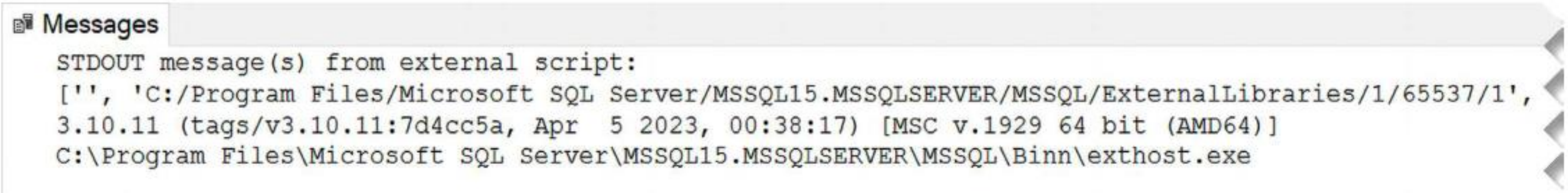


	external_language_id	language	create_date	principal_id
1	1	R	2023-09-16 12:48:44.800	4
2	2	Python	2023-09-16 12:48:44.803	4
3	65537	py310	2023-09-16 13:24:34.830	1
4	65539	r422	2023-09-17 13:08:32.377	1

Your First Python Script from SQL Server

- The tool in T-SQL to execute a script is the system-stored procedure sp_execute_external_script
- Essential parameters
  - External language identifier (e.g. *py310*)
  - String of the script to execute

```
EXEC sp_execute_external_script @language = N'py310',
@script=N'
import sys
print(sys.path)
print(sys.version)
print(sys.executable)';
```



```
STDOUT message(s) from external script:
['', 'C:/Program Files/Microsoft SQL Server/MSSQL15.MSSQLSERVER/MSSQL/ExternalLibraries/1/65537/1',
3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)]
C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Binn\exthost.exe
```

A closer look at `sp_execute_external_script`

DEMO 01

Understanding Input and Output parameters of `sp_execute_external_script`

Implementing the predictive backend

Using FLAML AutoML Model in SQL Server

Simplified ML Model Creation by Analysts (out-of-scope)

- FLAML AutoML makes it easy to create machine learning models
- Models can be serialized and saved in pickle format for reuse

Set Up Model for SQL Server

- Place the pickle model in a folder accessible by SQL Server
- Save the model in binary format within a database table

Prediction Integration

- Use the model for predictions
- Return predicted labels and probabilities based on input parameters
- All into a stored procedure

DEMO 02

Implementing the predictive backend

Let’s Use SQL Server External Languages with Power BI

Why the Need of External Languages with Power BI

- Enterprise Architecture Constraints
  - Power BI’s Personal Gateway mode for built-in Python and R is unsuitable for enterprise-level, multi-user environments
- Privacy and Security Challenges
  - Public privacy levels in Power BI can expose sensitive data when combining data sources
- Processing Large SQL Server Datasets
  - External languages leverage SQL Server's resources to handle large datasets securely and efficiently
- Missing Libraries in Power BI Service
  - Overcomes the limited library support in Power BI by running scripts directly in SQL Server
- Key Benefit
  - Keeps advanced analytics workflows scalable, secure, and integrated into the main data flow

What We’ll Do in Power BI

Objective

- Enable real-time prediction updates in Power BI using the Titanic Survival model without updating the dataset

Solution Overview

- Use DirectQuery mode to query SQL Server in real-time
- SQL Server runs the model scoring script and enriches data dynamically

Key Implementation Steps

- Call `sp_execute_external_script` within a stored procedure
- Use stored procedures for data enrichment in Power BI

Execute Stored Procedures in DirectQuery

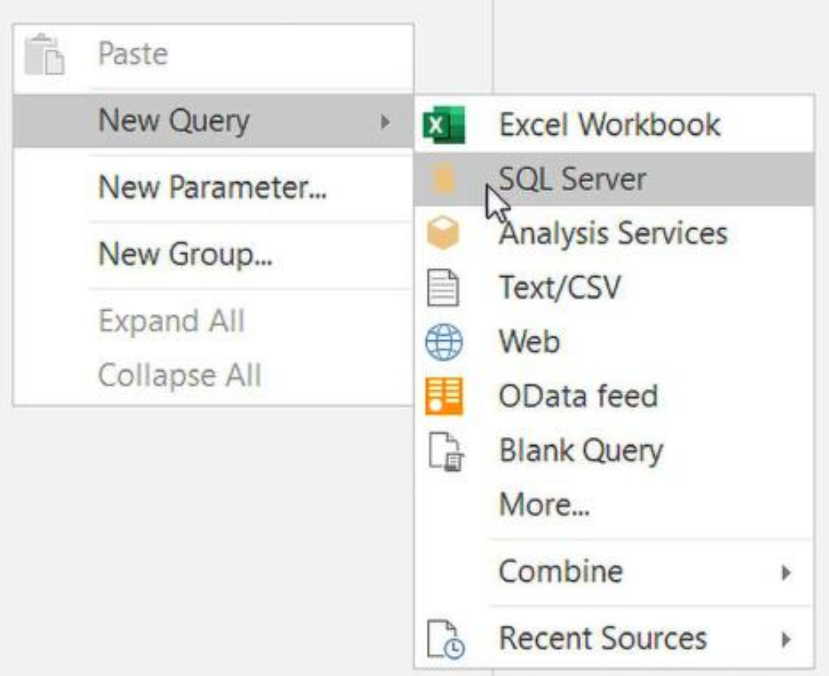
Here what I tried do

1. Enter an `EXEC <my-stored-procedure>` … statement in a DirectQuery SQL Server new Query in Power Query
   - Once entered, all works fine, and you can see the data as output
2. When you click *Close & Apply*, you get the mysterious *Incorrect syntax near the keyword 'EXEC'* error

What happens behind the scene?

- I used the SQL Server Profiler and I got this metadata validation query:

```
select * from (
    EXEC <my-stored-procedure> … WITH RESULT
SETS ((…))
) SourceQuery where 1 = 2
```



Converting an EXEC to SELECT ... FROM 1/2

- In SQL Server, external data can be retrieved using the OPENROWSET function with OLE DB providers:

```
SELECT *
FROM OPENROWSET('SQLNCLI', 'server=.;Trusted_Connection=yes;',
        'SELECT configuration_id, name FROM sys.configurations’)
```

- The SQLNCLI OLE DB provider for connecting to SQL Server is pre-installed with the SQL Server setup.
- The first time you run the above query, you get an error:

Msg 15281, Level 16, State 1

*SQL Server blocked access to STATEMENT 'OpenRowset/OpenDatasource' of component '**Ad Hoc Distributed Queries**' because this component is turned off as part of the security configuration for this server.*

Converting an EXEC to SELECT ... FROM 2/2

- Ad Hoc Distributed Queries feature must be enabled:

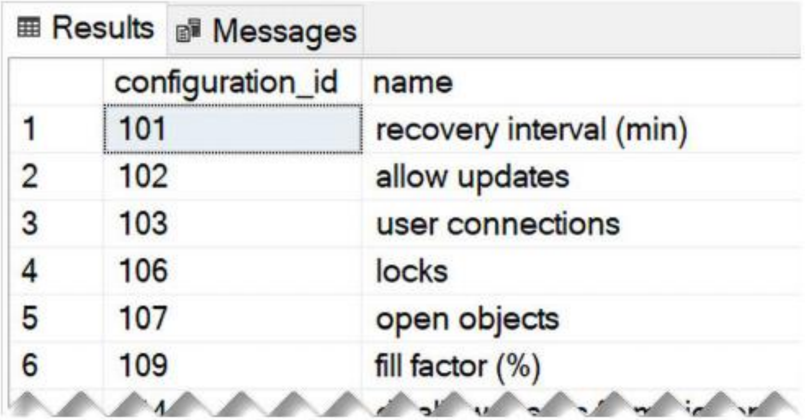
```
-- To show advanced options
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;

-- To enable Ad Hoc Distributed Queries
EXEC sp_configure 'Ad Hoc Distributed Queries', 1;
RECONFIGURE;
```

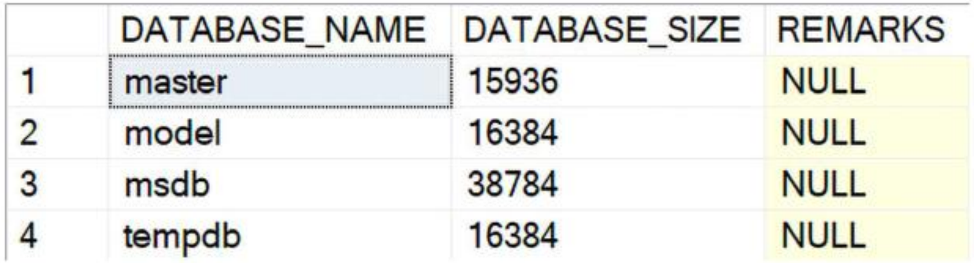
- Now the query works fine!
- But what if we try to run an EXEC query through the OPENROWSET?

```
SELECT *
FROM OPENROWSET('SQLNCLI',
'server=.;Trusted_Connection=yes;',
                 'EXEC [sys].[sp_databases]')
```

- It works too! ☺



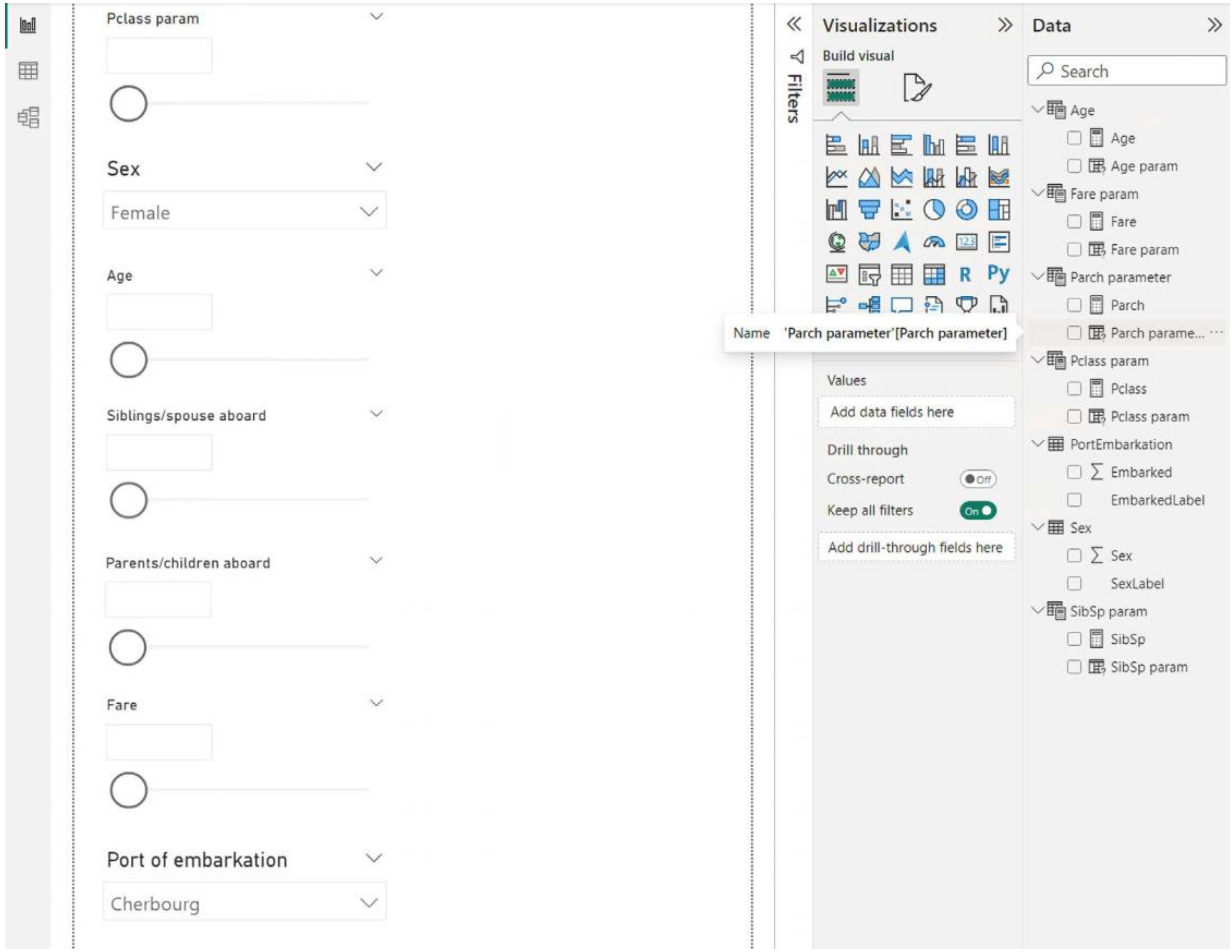
	configuration_id	name
1	101	recovery interval (min)
2	102	allow updates
3	103	user connections
4	106	locks
5	107	open objects
6	109	fill factor (%)



	DATABASE_NAME	DATABASE_SIZE	REMARKS
1	master	15936	NULL
2	model	16384	NULL
3	msdb	38784	NULL
4	tempdb	16384	NULL

Let’s Put it All Into Practice in Power BI

The Report Backbone



Sex: Female

Port of embarkation: Cherbourg

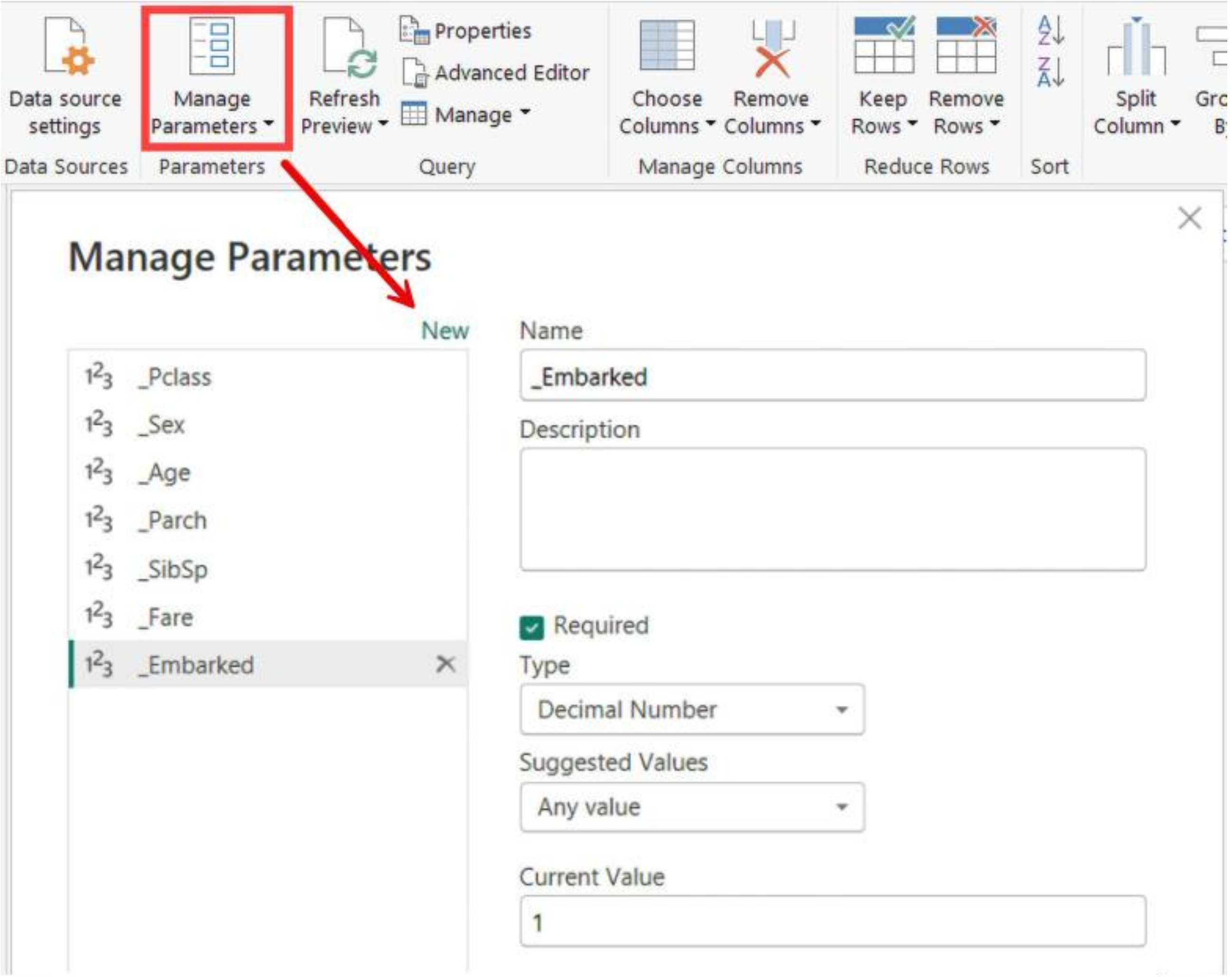
Name 'Parch parameter'[Parch parameter]

Cross-report: Off

Keep all filters: On

- Each slider shows data from the respective table
- Each slider is related to a ML model parameter
- DirectQuery call to the stored procedure is done into Power Query, so you need Parameters to map sliders values to the stored procedure parameters

Define Parameters in Power Query



Manage Parameters

New

- _Pclass
- _Sex
- _Age
- _Parch
- _SibSp
- _Fare
- _Embarked

Name: _Embarked

Description:

☑ Required

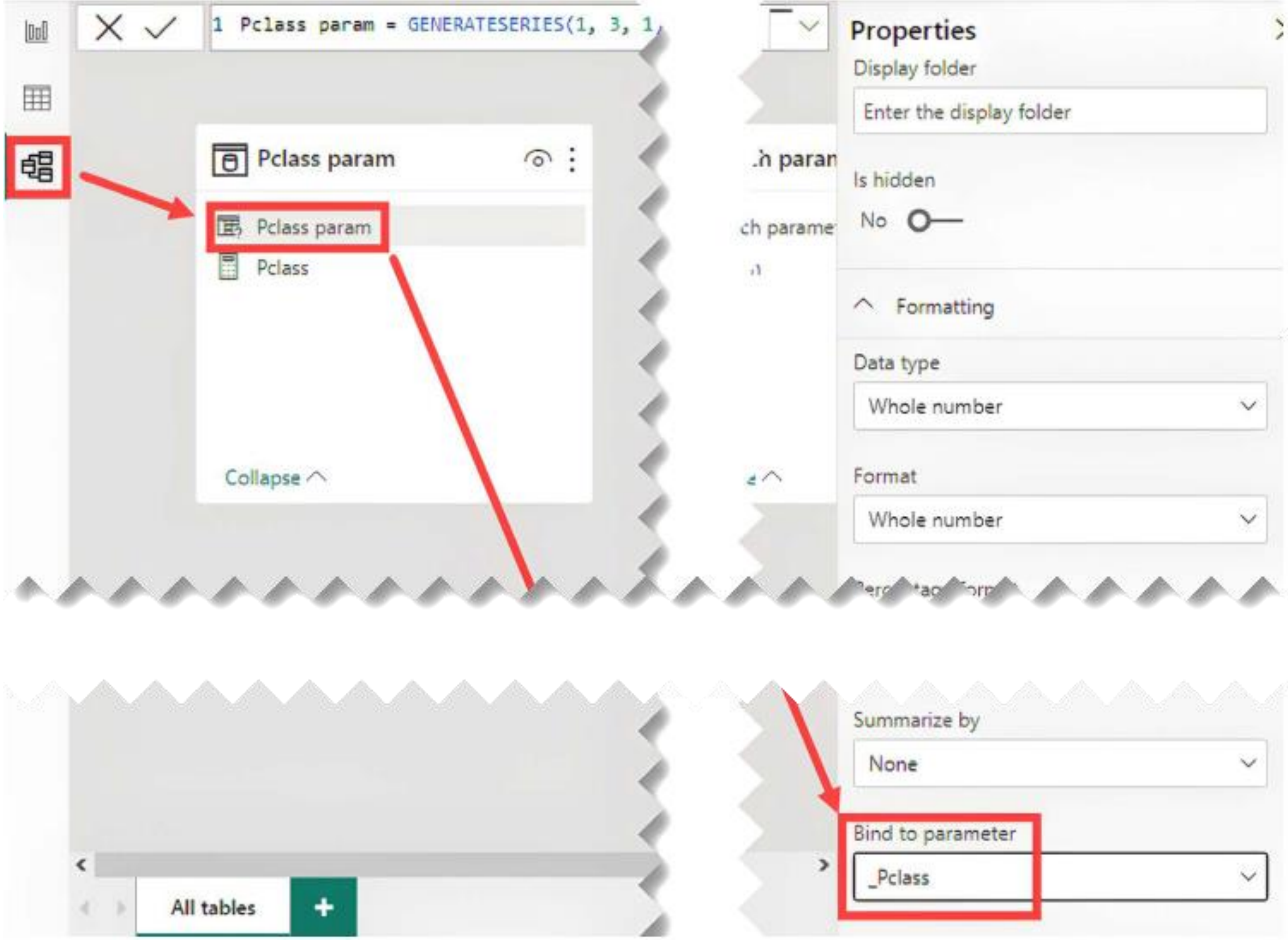
Type: Decimal Number

Suggested Values: Any value

Current Value: 1

- Set the correct data type for each parameter according to the ones expected by the stored procedure
- You can leave 1 as *Current Value* for all of them

Bind Parameters to Table Fields



```
1 Pclass param = GENERATESERIES(1, 3, 1,
```

Properties

Display folder: Enter the display folder

Is hidden: No

Formatting

Data type: Whole number

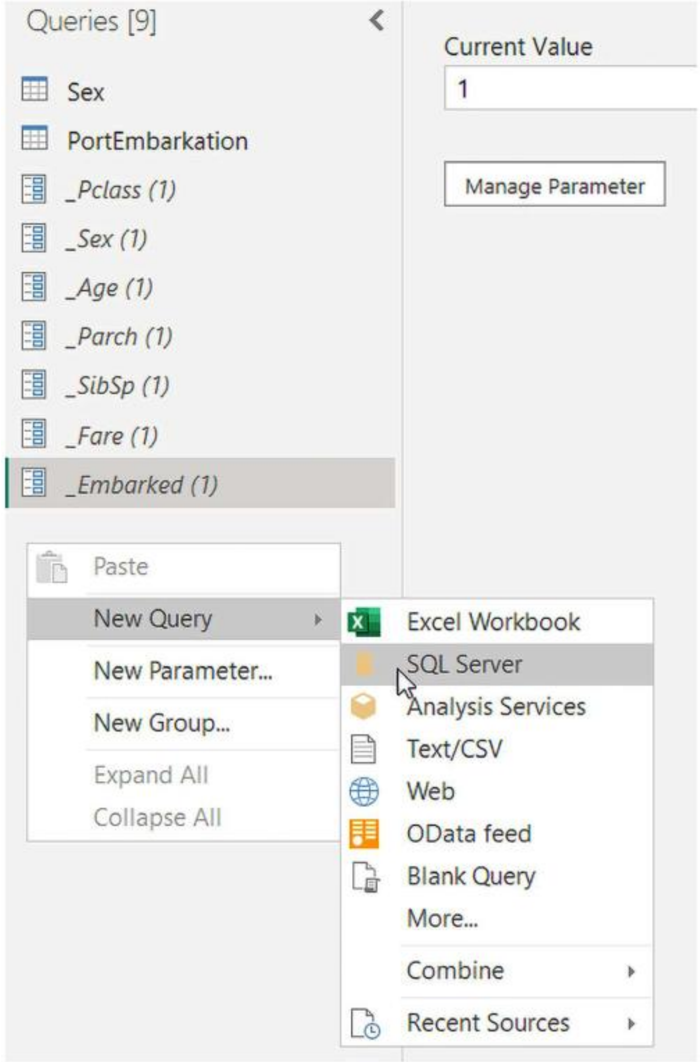
Format: Whole number

Summarize by: None

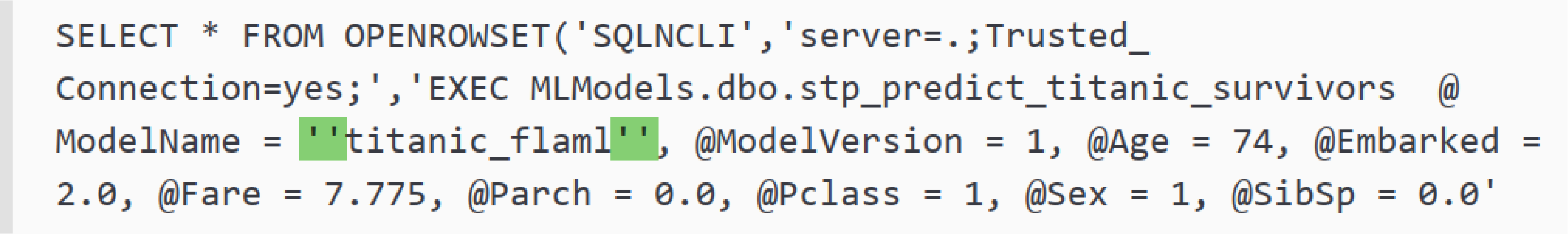
Bind to parameter: _Pclass

- Go to *Model View* and select the numeric field to bind to a parameter
- In its *Advanced* properties set the Parameter name into the *Bind to parameter* combo box
- Select None for *Summarize by*

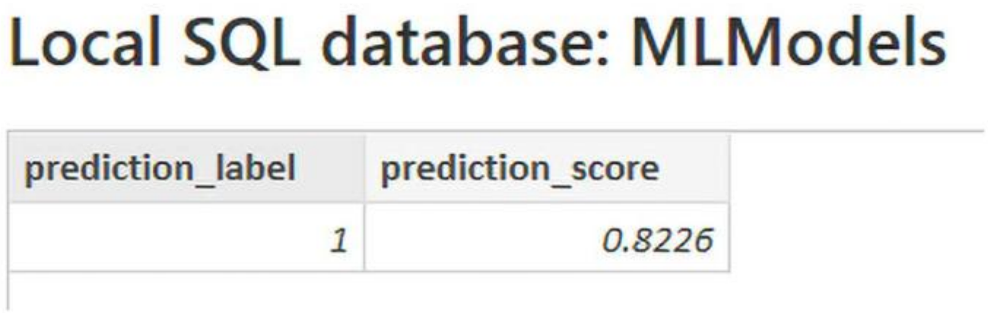
Create the DirectQuery to SQL Server



SQL Statement with the trick and fixed parameters:



```
SELECT * FROM OPENROWSET('SQLNCLI','server=.;Trusted_
Connection=yes;','EXEC MLModels.dbo.stp_predict_titanic_survivors  @
ModelName = ''titanic_flaml'', @ModelVersion = 1, @Age = 74, @Embarked =
2.0, @Fare = 7.775, @Parch = 0.0, @Pclass = 1, @Sex = 1, @SibSp = 0.0'
```

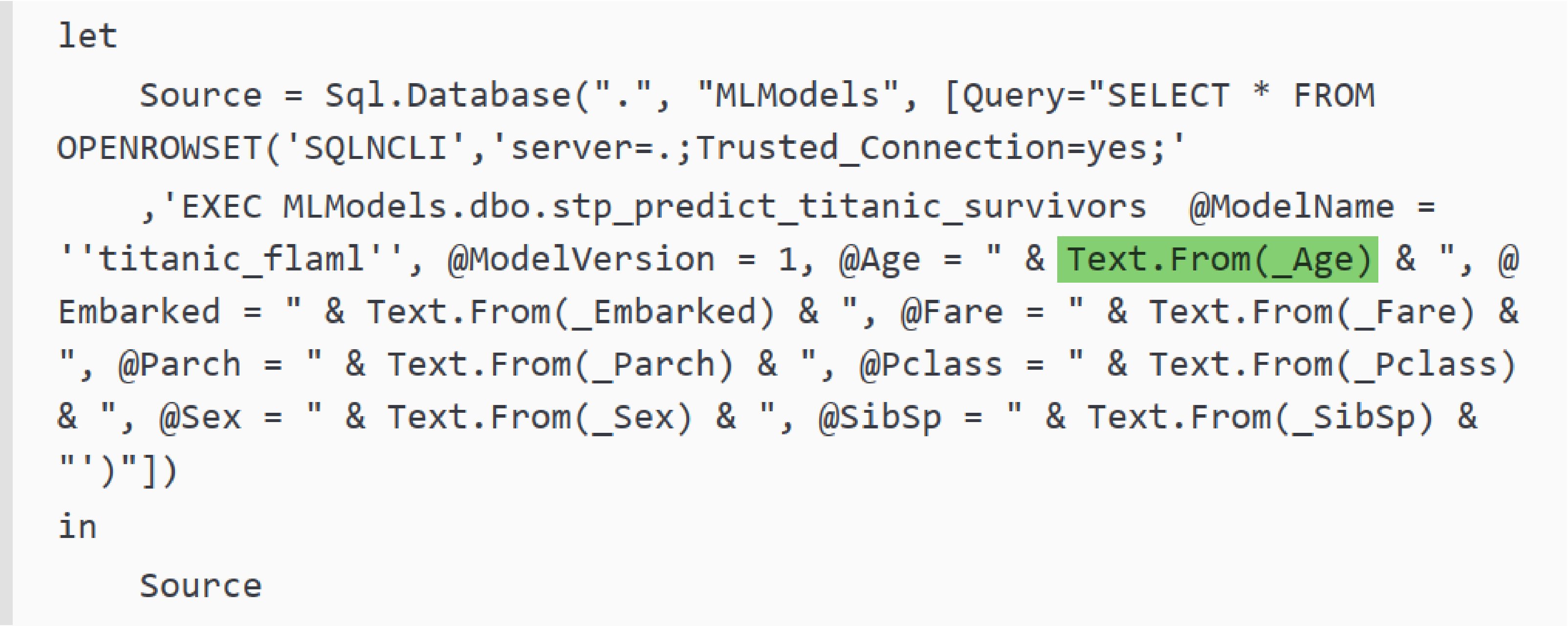


Local SQL database: MLModels

prediction_label	prediction_score
1	*0.8226*

Let’s Use Parameters in SQL Statement

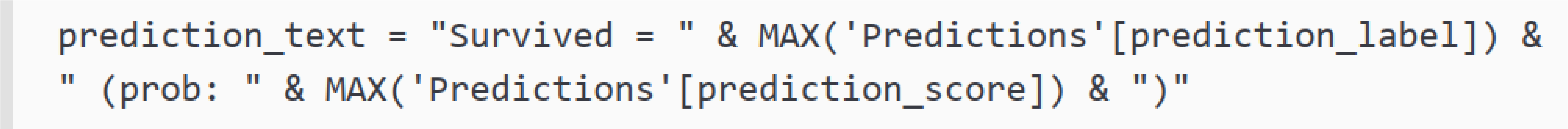
Replace constants with `Text.From(…)` in the M expression



```
let
    Source = Sql.Database(".", "MLModels", [Query="SELECT * FROM
OPENROWSET('SQLNCLI','server=.;Trusted_Connection=yes;'
    ,'EXEC MLModels.dbo.stp_predict_titanic_survivors  @ModelName =
''titanic_flaml'', @ModelVersion = 1, @Age = " & Text.From(_Age) & ", @
Embarked = " & Text.From(_Embarked) & ", @Fare = " & Text.From(_Fare) &
", @Parch = " & Text.From(_Parch) & ", @Pclass = " & Text.From(_Pclass)
& ", @Sex = " & Text.From(_Sex) & ", @SibSp = " & Text.From(_SibSp) &
"')"])
in
    Source
```

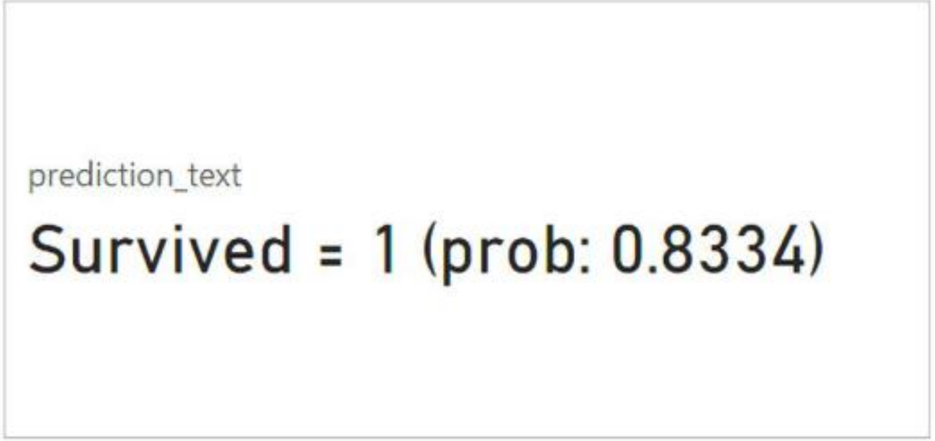
Show the Prediction in a Card Visual

Create a new DAX measure to combine prediction and probabilities…



```
prediction_text = "Survived = " & MAX('Predictions'[prediction_label]) &
" (prob: " & MAX('Predictions'[prediction_score]) & ")"
```

… and show it in your Card!



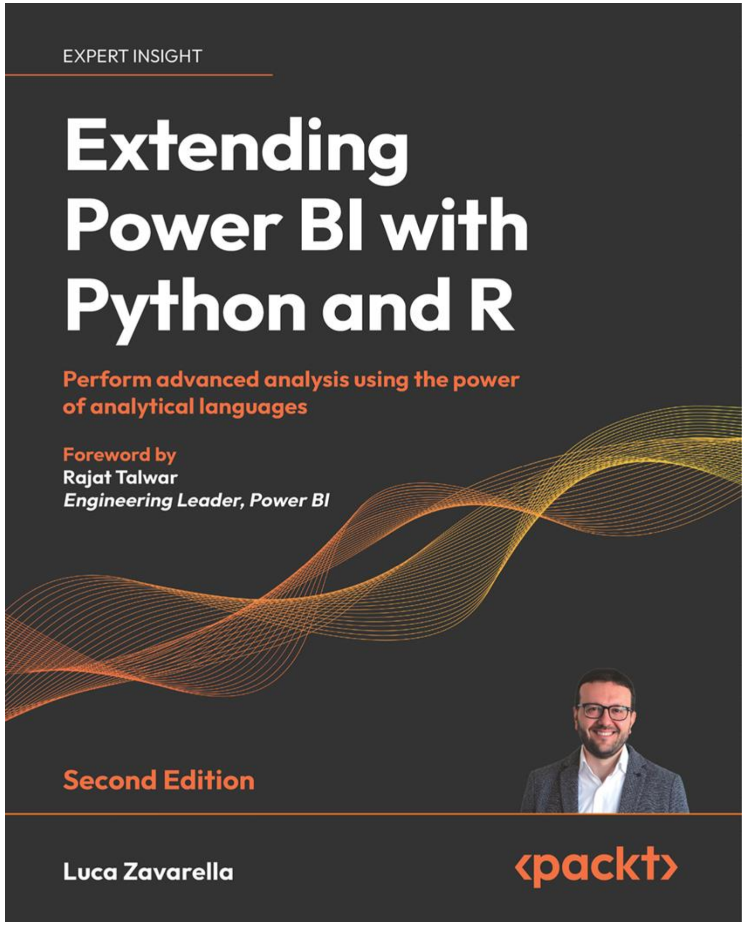
prediction_text

Survived = 1 (prob: 0.8334)

DEMO 03

Let’s Play With the Power BI Predictive Report

Learn more about these topics



EXPERT INSIGHT

Extending Power BI with Python and R

Perform advanced analysis using the power of analytical languages

Foreword by
Rajat Talwar
Engineering Leader, Power BI

Second Edition

Luca Zavarella

‹packt›

All steps shown in this session are detailed in my book!

Paperback & Kindle

- Amazon.it
- Amazon.com
- Amazon.<what-you-want>

PDF, EPUB, MOBI (DRM free)

- Packtpub.com

References

- Install a Python custom runtime for SQL Server 2019 (https://learn.microsoft.com/en-us/sql/machine-learning/install/custom-runtime-python)
- Install SQL Server 2022 Machine Learning Services (Python and R) on Windows (https://learn.microsoft.com/en-us/sql/machine-learning/install/sql-machine-learning-services-windows-install-sql-2022)
- Fix: Getting R and Python to actually work on SQL Server 2022 (https://blog.greglow.com/2024/03/04/fix-getting-r-and-python-to-actually-work-on-sql-server-2022/)
- SQL Server Language Extension Releases (https://github.com/microsoft/sql-server-language-extensions/releases)
- SQL Server on Windows: Isolation changes for Machine Learning Services (https://learn.microsoft.com/en-us/sql/machine-learning/install/sql-server-machine-learning-services-2019)
- Implied authentication (loopback requests) (https://learn.microsoft.com/en-us/sql/machine-learning/concepts/security?view=sql-server-ver16#implied-authentication-loopback-requests-1)
- Manage Python and R workloads with Resource Governor in SQL Server Machine Learning Services (https://learn.microsoft.com/en-us/sql/machine-learning/administration/resource-governor)

Thank you!!!