SUMMIT2017

PASS

Building One Million Predictions Per Second Using SQL-R

Accelerate insights from your DATA

Amit Banerjee (@banerjeeamit)
Senior Program Manager
Microsoft Database Systems

C:\Users\> whoami

An affair with SQL Server for nearly a decade

Was part of SQL Escalation Services and Premier Field Engineering team at Microsoft

Now a Sr. Program Manager on the Microsoft Database Systems team focusing on performance, scale, HADR and data movement

Speaker at SQL PASS 24HOP TechEd Virtual TechDays User Groups SQL Saturdays

Dabble around with supportability tools and have contributed to SQL Backup Simulator SQLDIAG/PSSDIAG Manager and SQL Nexus

Co-authored “Professional SQL Server 2012: Internals and Troubleshooting” and “Pro SQL Server on Azure”

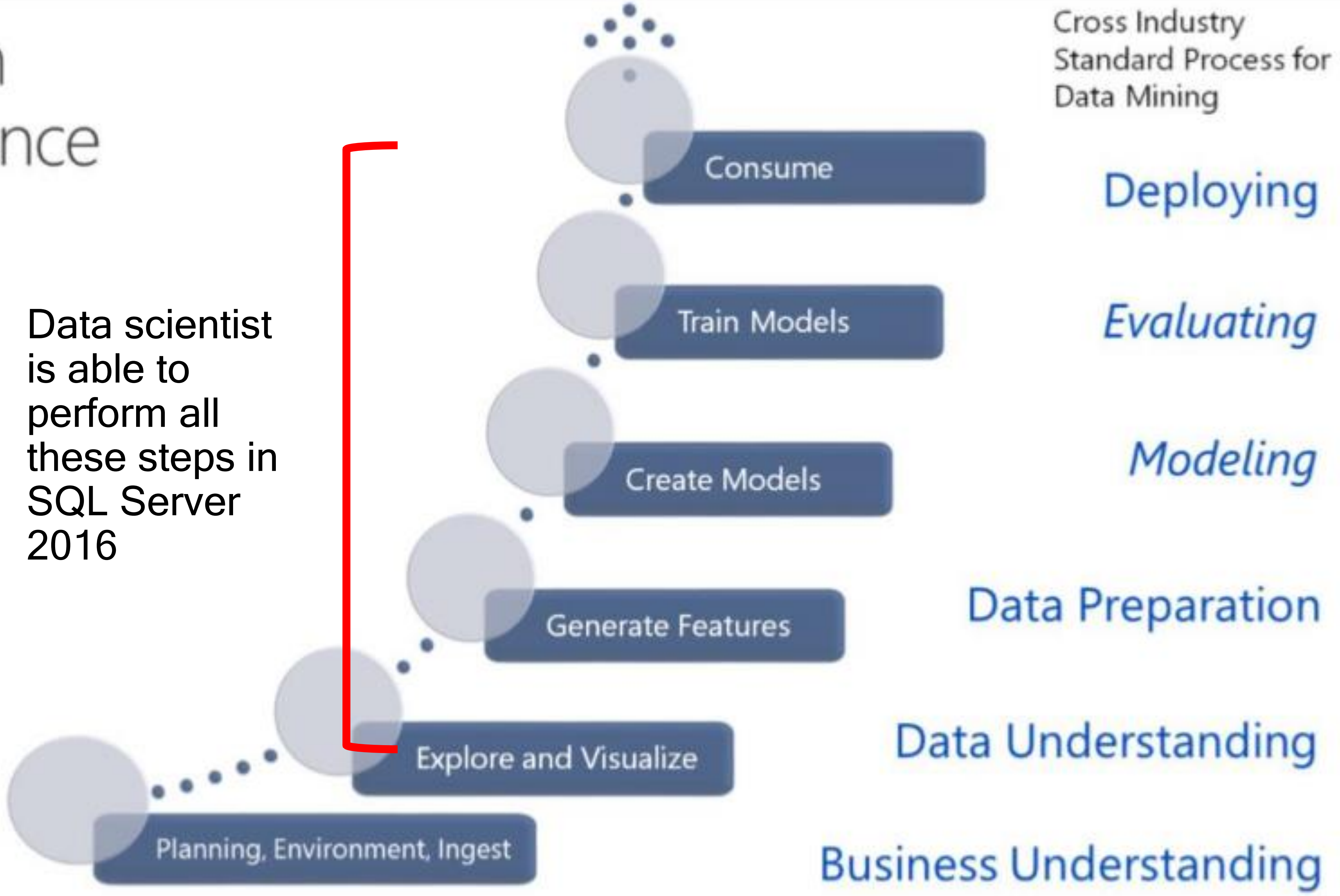
Own TroubleshootingSQL.com

Agenda

- Data Science Process
- Bringing Analytics to Data
- Demo
- Optimization Tips

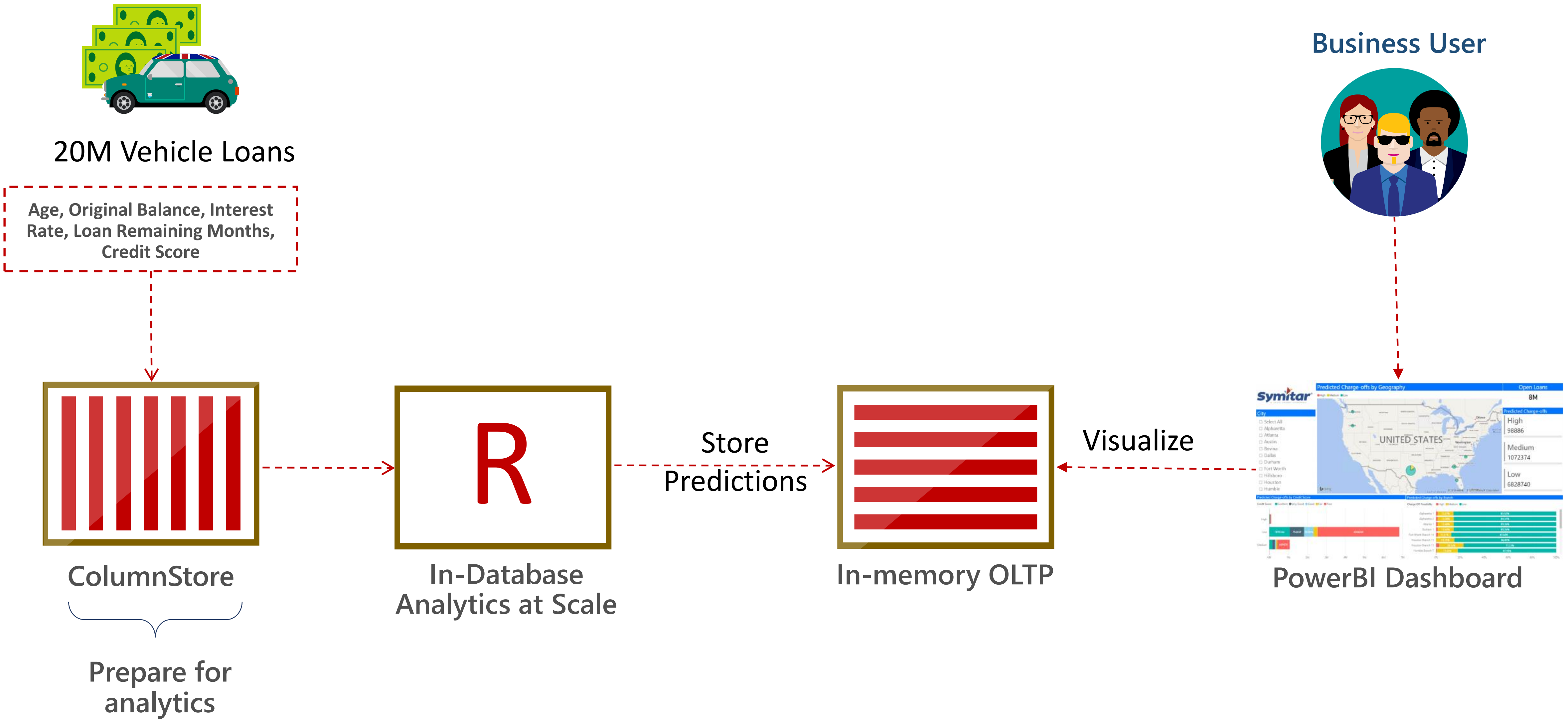
The Team Data Science Process

Data scientist is able to perform all these steps in SQL Server 2016

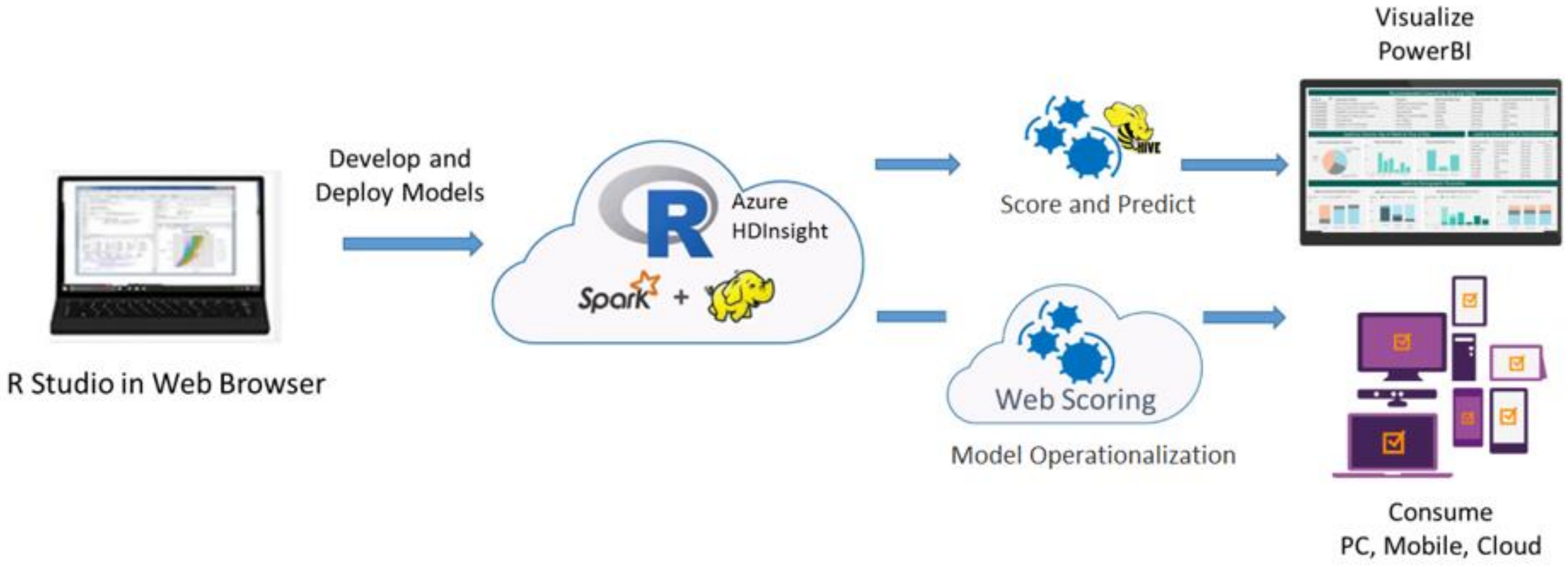


Jack Henry

A leading provider for banking solutions for credit unions across Americas



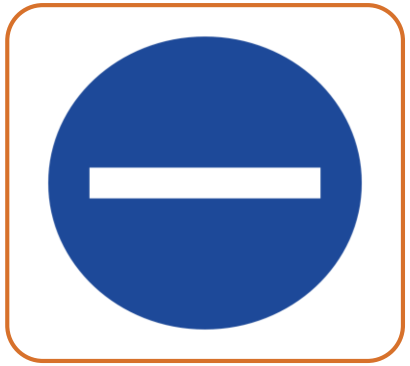
Deploying Loan Charge-off Prediction using HDInsight



Using Machine Learning Services in SQL Server

Bringing Analytics to the Data

- Data already in SQL
- Use T-SQL know-hows to do ETL
- Use the power of in-memory OLTP and column store indexing to enhance speed of ETL
- RevoScaleR package to provide parallelism and scale



Making the data travel

- Data sources not in SQL
- Data sinks not in SQL
- Complex ETL needed
- Long running R script

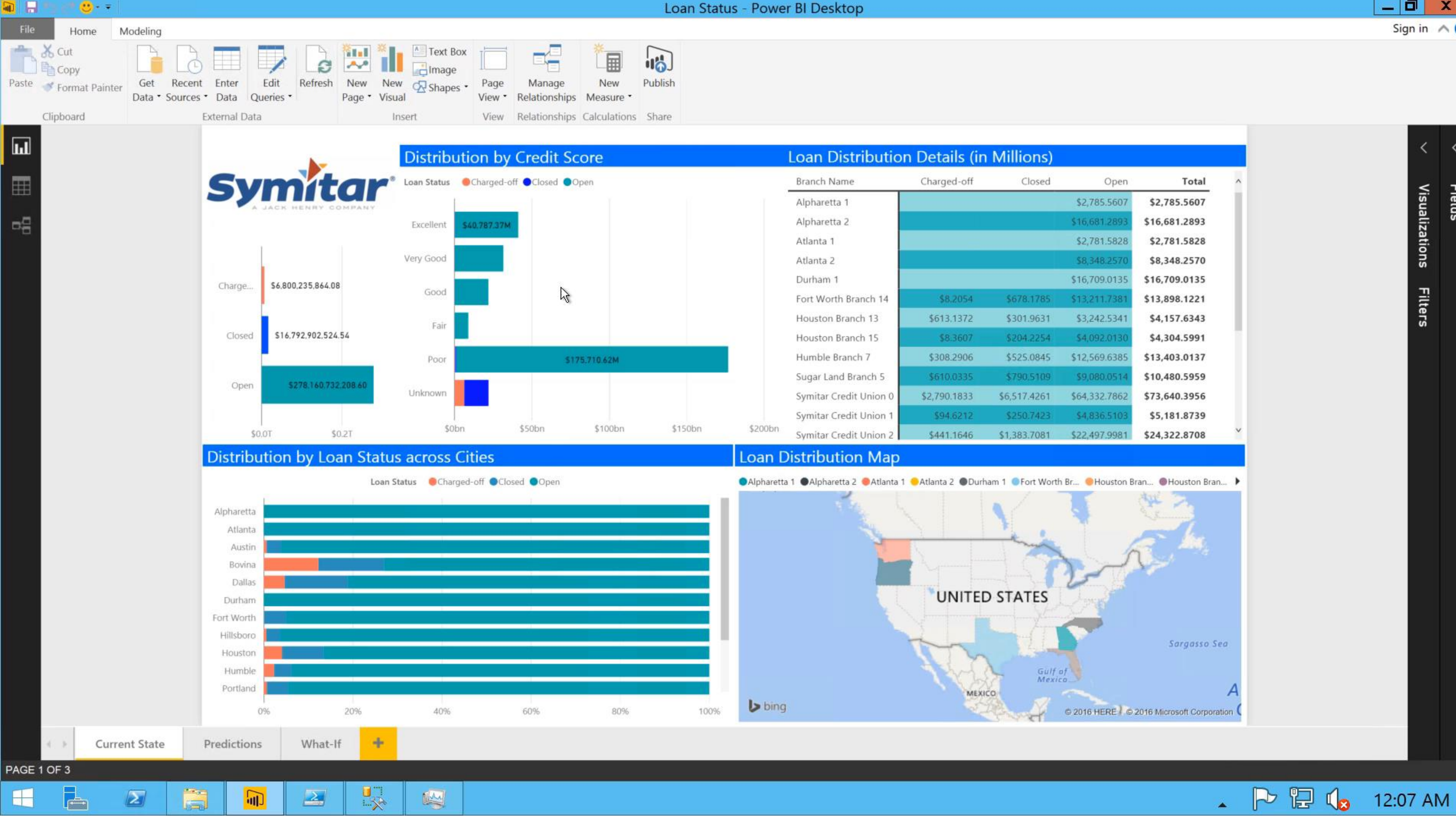
Using Python in SQL Server 2017

- Elimination of data movement
- Easy deployment
- Enterprise-grade performance and scale
- Rich extensibility
- Wide availability at no additional costs

sp_execute_external

```
sp_execute_external_script

    @language = N'language' ,
    @script = N'script',

    @input_data_1 = ] 'input_data_1'
    [ , @input_data_1_name = ] N'input_data_1_name' ]
    [ , @output_data_1_name = 'output_data_1_name' ]
    [ , @parallel = 0 | 1 ] [ , @params = ]

    N'@parameter_name data_type [ OUT | OUTPUT ] [ ,...n ]'
    [ , @parameter1 = ] 'value1' [ OUT | OUTPUT ] [ ,...n ]
    [ WITH <execute_option> ]
[;]

<execute_option>::=
{
    { RESULT SETS UNDEFINED }
    | { RESULT SETS NONE }
    | { RESULT SETS ( <result_sets_definition> ) }
 }

<result_sets_definition> ::=
{
  (
     { column_name
       data_type
     [ COLLATE collation_name ]
     [ NULL | NOT NULL ] }
     [,...n ]
  )
  | AS OBJECT
     [ db_name . [ schema_name ] . | schema_name . ]
     {table_name | view_name | table_valued_function_name }
  | AS TYPE [ schema_name.]table_type_name
}
```

```
EXEC sp_execute_external_script

    @language = N'R'
    , @script = N'iris_data <- iris;'
    , @input_data_1 = N''
    , @output_data_1_name = N'iris_data'
    WITH RESULT SETS (("Sepal.Length" float not null,
       "Sepal.Width" float not null,
       "Petal.Length" float not null,
       "Petal.Width" float not null, "Species" varchar(100)));
END;

go
```



Branch Name	Charged-off	Closed	Open	Total
Alpharetta 1			$2,785.5607	$2,785.5607
Alpharetta 2			$16,681.2893	$16,681.2893
Atlanta 1			$2,781.5828	$2,781.5828
Atlanta 2			$8,348.2570	$8,348.2570
Durham 1			$16,709.0135	$16,709.0135
Fort Worth Branch 14	$8.2054	$678.1785	$13,211.7381	$13,898.1221
Houston Branch 13	$613.1372	$301.9631	$3,242.5341	$4,157.6343
Houston Branch 15	$8.3607	$204.2254	$4,092.0130	$4,304.5991
Humble Branch 7	$308.2906	$525.0845	$12,569.6385	$13,403.0137
Sugar Land Branch 5	$610.0335	$790.5109	$9,080.0514	$10,480.5959
Symitar Credit Union 0	$2,790.1833	$6,517.4261	$64,332.7862	$73,640.3956
Symitar Credit Union 1	$94.6212	$250.7423	$4,836.5103	$5,181.8739
Symitar Credit Union 2	$441.1646	$1,383.7081	$22,497.9981	$24,322.8708

Building a SQL Server scoring engine

- **Fast Models**
- Correc
- Fast
- Fast Reads
- Uniform resource usage

Use R Studio to design your models or you could even use T-SQL to design your models and test

Building a SQL Server scoring engine

- Fast Models
- Correct attribute selection
- Fast Inge
- Fast Rea
- Uniform resource usage

Important to select the right attributes to ensure that the correct attributes are selected

Building a SQL Server scoring engine

- Fast Models
- Correct attribute selection
- **Fast Ingestion**
- Fast Reads
- Uniform re

In-Memory OLTP can help here. Staging tables can be schema-only.

Building a SQL Server scoring engine

- Fast Models
- Corr
- Fast In
- Fast Reads
- Uniform resource usage

Most of these models will read large datasets. The I/O can be reduced which tends to be slowest part using Columnstore indexes

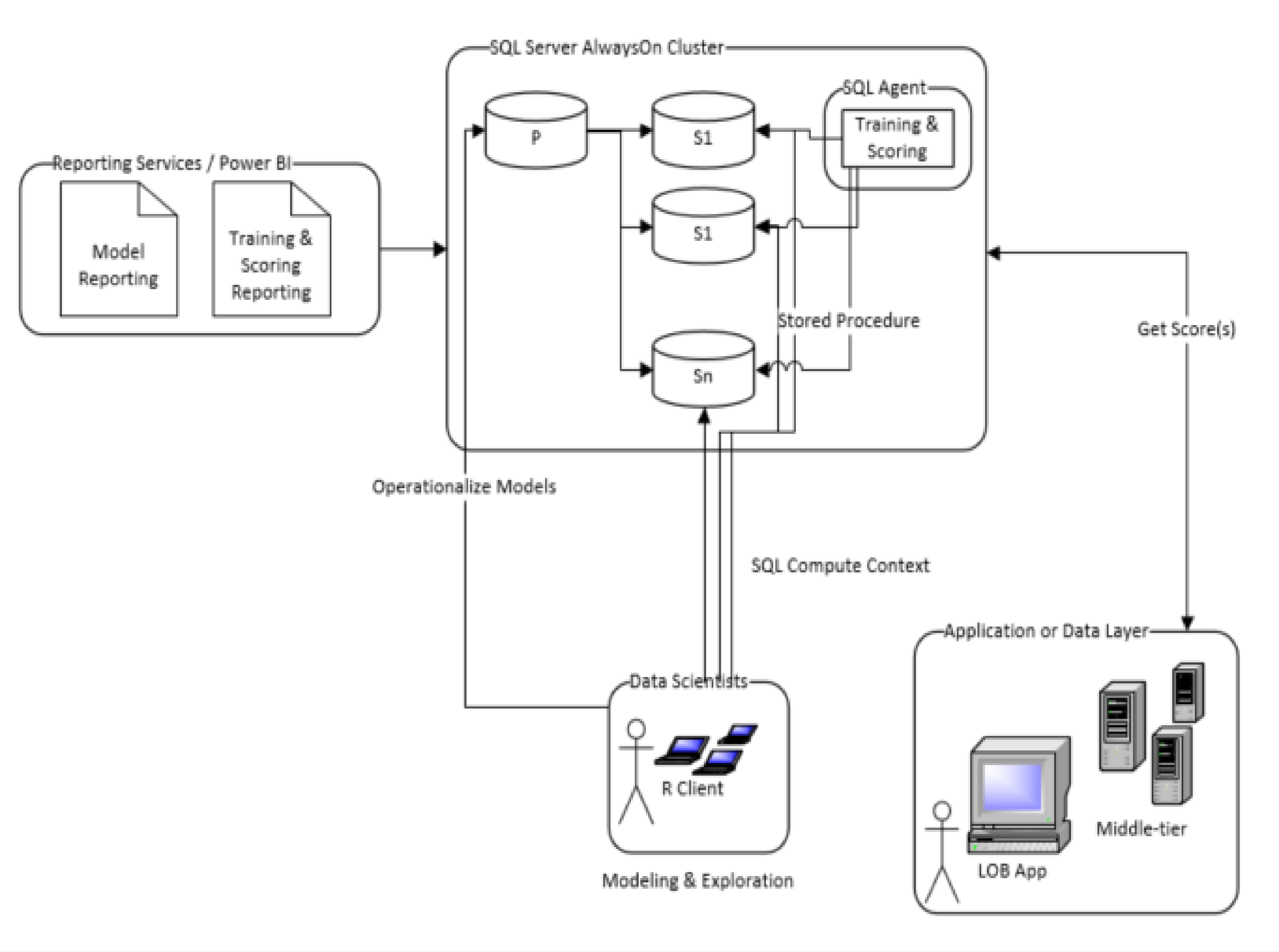
Building a SQL Server scoring engine

- Fast Models
- Cor
- Fast
- Fast Reads
- **Uniform resource usage**

You want all the CPUs to be used in parallel. This can be achieved with Resource Governor group assignment for the available CPUs/NUMA nodes. Pick the right CPU optimized machines

Deployment Using:

- Triggers
- Powershell scripts
- SQL agent jobs



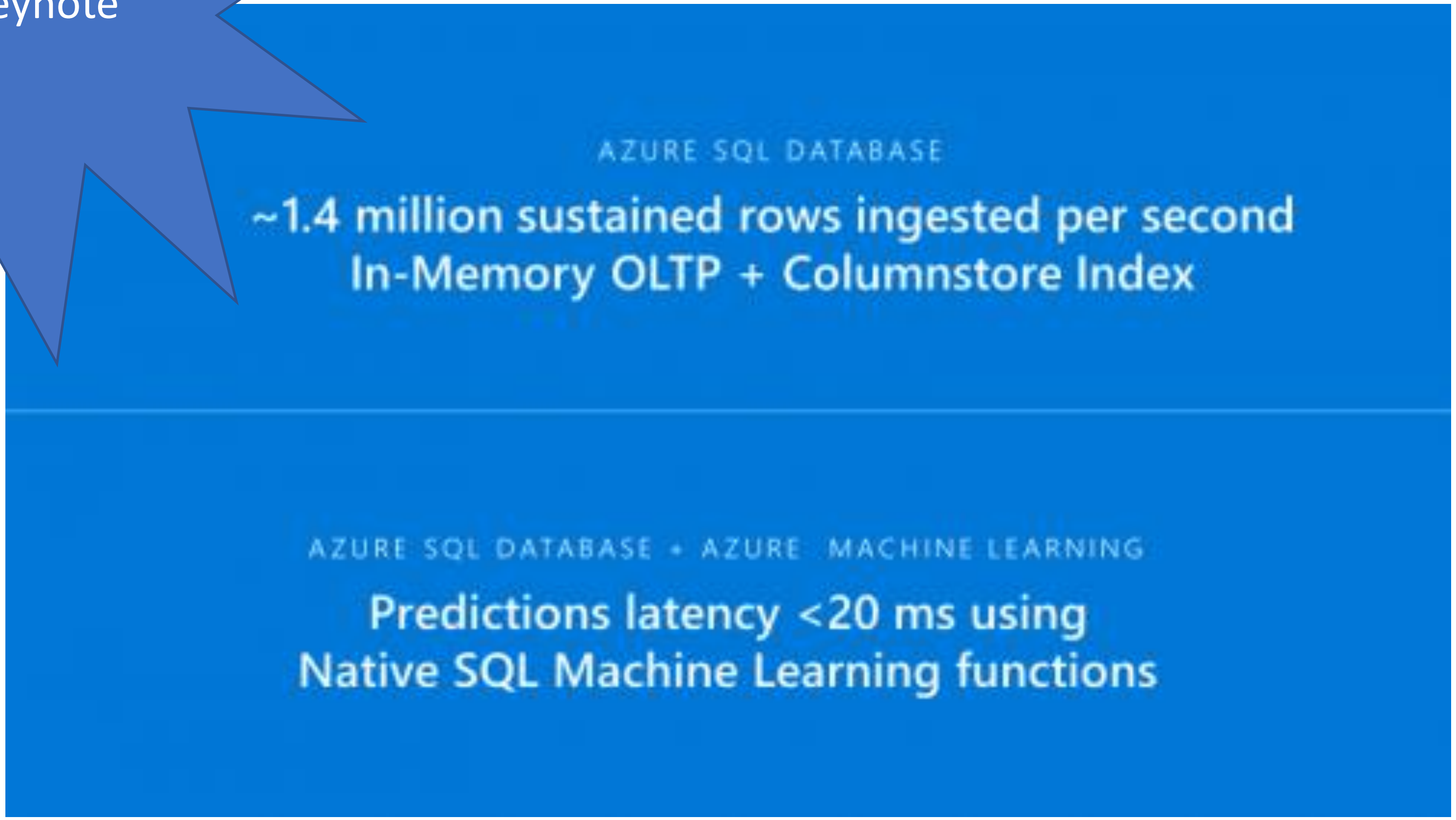
Do you want to use this?

A fully functional solution that can work on using SQL Server 2016 utilizing in-memory OLTP, columnstore indexes and SQL-R services

A fully functional solution that can work on using HDInsight Spark with R-services

Cortana Intelligence Quickstart Gallery

Rohan’s Keynote



Cortana Intelligence Solutions PREVIEW DEPLOYMENTS SOLUTIONS GALLERY

1 | Create new deployment

2 | Provide configuration parameters

3 | Resource provisioning (automated)

4 | Done

Loan ChargeOff Prediction with SQL Server

Prerequisites
You need to accept the Terms of Use of the Data Science Virtual Machine on your Azure Subscription before you deploy this VM the first time **by clicking here**.

Estimated Provisioning Time: **20 Minutes**

Not ready to deploy or need more information on Cortana Intelligence Solutions? Contact us.

Deployment name

(Deployment name must be between 3 and 9 characters, start with a lowercase letter, and contain only lowercase letters and numbers.)

Subscription

(0ffa90b2-4a7a-4952-9ca5-bbfd7d437d0f)

Location

East US

Description (optional)

License

Cancel Create

Cortana Intelligence Solutions PREVIEW DEPLOYMENTS SOLUTIONS GALLERY

1 | Create new deployment

2 | **Provide configuration parameters**

3 | Resource provisioning (automated)

4 | Done

Solution: Loan ChargeOff Prediction with SQL Server

Resource group: tigersqlr

Status: **Action required**

Username (Windows Virtual Machine)

tiger

Password (Windows Virtual Machine)

●●●●●●●●●●●●

Name for the Virtual Machine.

tigersqldemor

Windows computer name cannot be more than 15 characters long, be entirely numeric, or contain non-ASCII or special characters.

Size for the Virtual Machine.

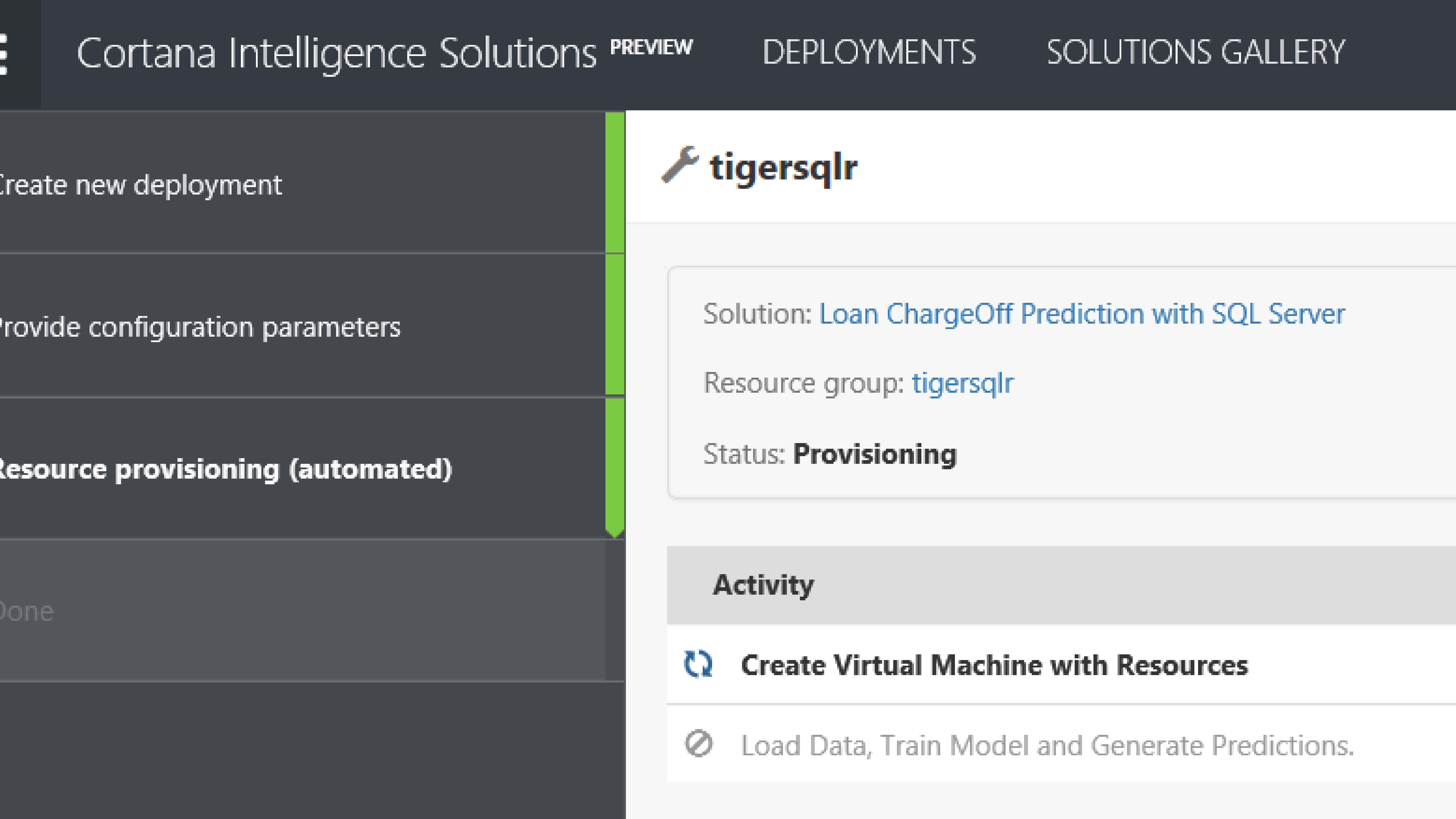
Standard_DS14_v2

Username (SQL Server)

tiger

Password (SQL Server)

●●●●●●●●●


Cortana Intelligence Solutions PREVIEW
DEPLOYMENTS
SOLUTIONS GALLERY
Create new deployment
Provide configuration parameters
Resource provisioning (automated)
Done
tigersqlr
Solution: Loan ChargeOff Prediction with SQL Server
Resource group: tigersqlr
Status: Provisioning
Activity
Create Virtual Machine with Resources
Load Data, Train Model and Generate Predictions.

Lets see this in action

DATA PREPARATION → FEATURE SELECTION → DATA PARTITION → PREDICTIVE MODELLING → SCORING → MODEL EVALUATION

References

- [What's new in Machine Learning Services in SQL Server]
- [Loan Classification using SQL Server 2016 R Services]
- [A walkthrough of Loan Classification using SQL Server 2016 R Services]
  [Using MicrosoftML in SQL-Server]
- [GitHub SQL Server Samples]
- [Quick Start template using SQL Server]
- [Quick Start template using HDInsight]
- [Performance tuning for R in SQL Server]