Microsoft

Evolving your Business with Data, Advanced Analytics, and AI

Joe Sack, Principal Program Manager, Microsoft

Predicting Operating Room Capacity

Scenario

- **Healthcare company** with separate hospitals, uses a confederation model in which affiliate medical centers jointly borrow and purchase common services and information technology support, such as **operating room booking systems**
- **Last-minute operating room cancellations** result in rooms going unused, over-staffing and lengthy patient wait-lists.
- Customer uses a combination of SSAS, T-SQL calculations and a iterative Data Mining Model to predict operating room capacity based on past booking history
- This solution has been effective in reducing operating room “spoilage”, but the solution is higher in complexity than desired by the customer.

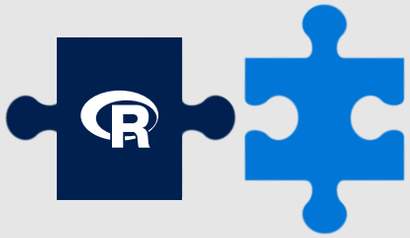
Solution

- Reduce complexity and reduce data movement by implementing the full modelling and prediction process within the SQL Server engine via SQL Server R Services

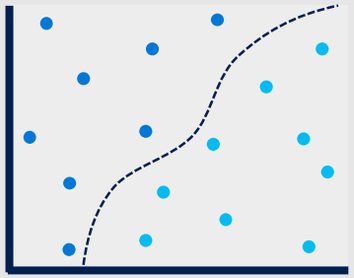
- Used rxLinMod function to locally train the model based on past booking history and predict future bookings behavior for future dates

Deploying predictive analytics

Develop, explore and experiment in your favorite IDE or language

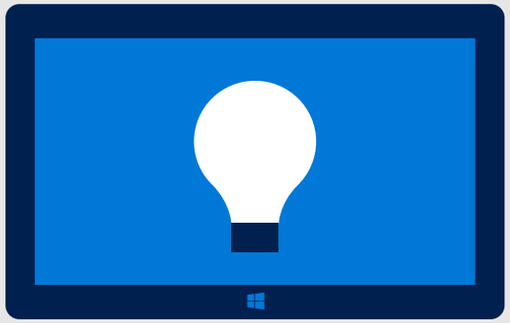


Train a model

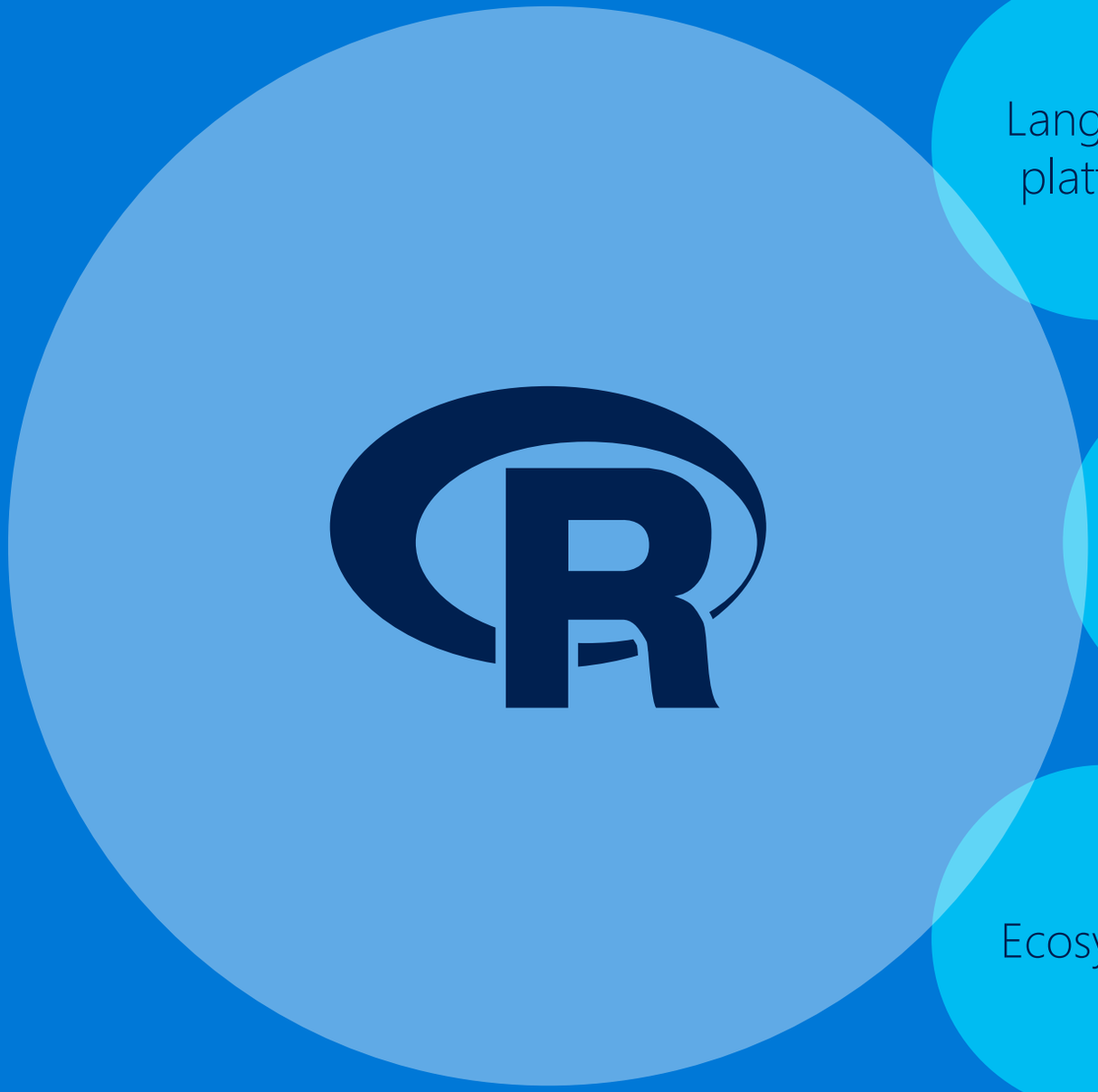


Deploy and predict with the model

Make your apps **intelligent** by **consuming** predictions



Develop | **Train** | **Deploy** | **Consume**



Language platform

Statistics programming language
Data visualization tool
Open source

Community

2.5+M users
Taught in most universities
Popular with new and recent grads
Thriving user groups worldwide

Ecosystem

10,000+ packages in CRAN
Scalable to big data
Rich application and platform integration

Challenges of using R

Data movement

- **Moving data from the DB to R**
- **Runtime becomes painful as data volumes grow**
- **Movement carries security risks**

Deployment

- **How do I call the R script from my production application?**

Scale and performance

- **Most R functions are single threaded and only accommodate datasets that fit into available memory**

Pain Points

Performance of data exploration activities may be slow (data locality | single-threaded | memory constraints)

Data may leave “trust boundary”

Solutions may not be easily operationalized, requiring re-coding in other solutions

Data Scientist may not be aware of already-cleansed and collected data

Data Scientist may be using data cleansing techniques which would be more efficiently performed by SQL Server

Database Engineer may be facilitating analytic and statistical operations which would be better performed using R

Microsoft Advanced Analytics Landscape

Azure Machine Learning

- Fully managed cloud service that enables you to easily build, deploy, and share predictive analytics solutions

SQL Server Machine Learning Services (In-Database)

- Supports both R and Python pushed SQL Server compute-context

Microsoft R Server

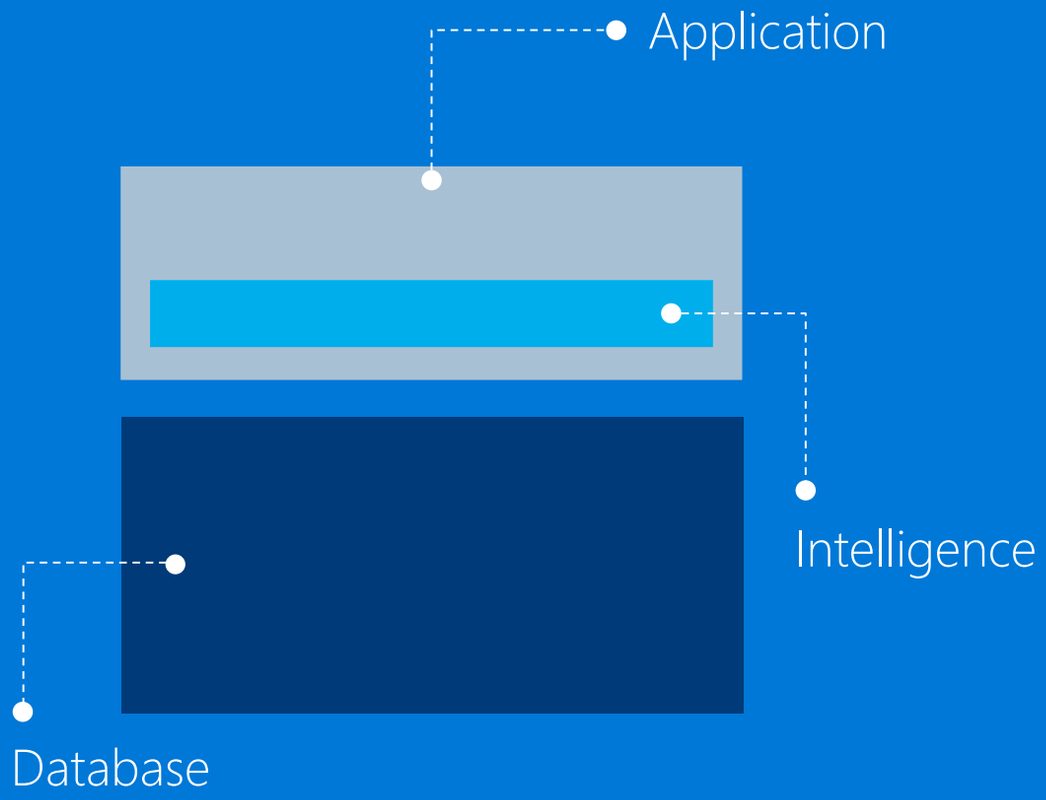
- For enterprise-level R deployments on Windows and Linux servers

Microsoft Machine Learning Server

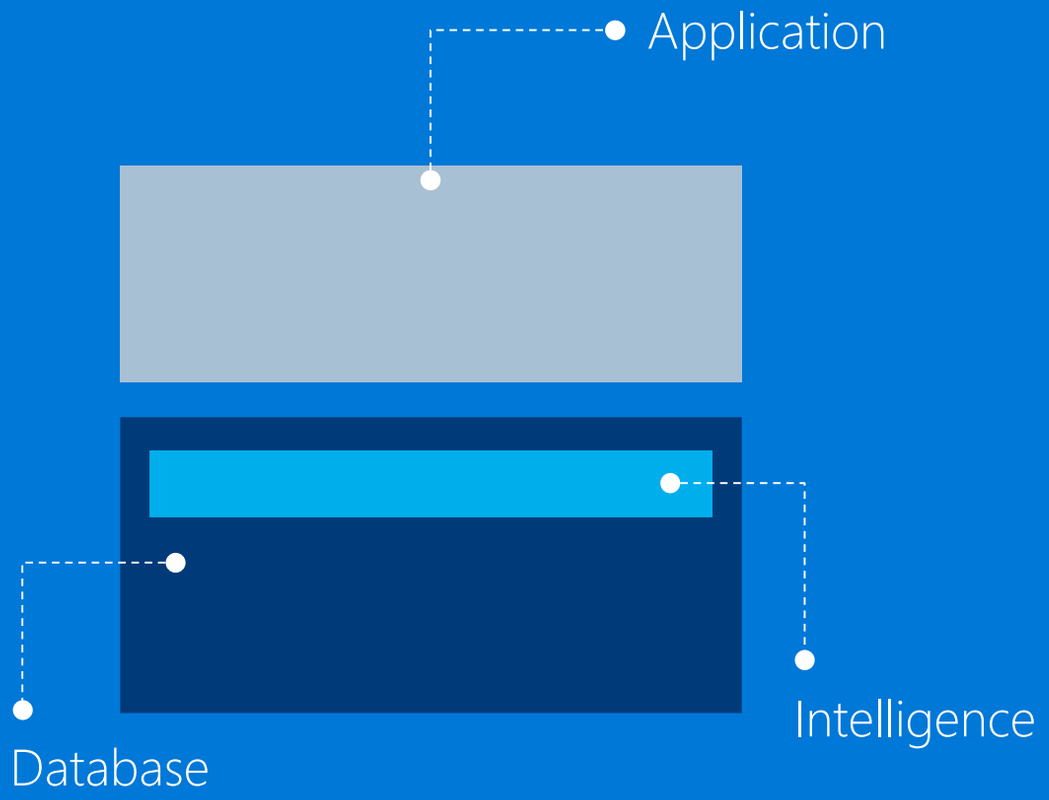
- Supports R and Python deployments on Windows servers, with expansion to other supported platforms planned for late 2017

In-database advanced analytics

Pushing intelligence to where data lives



Before



Intelligence built in to the DB

SQL Server ML Services solves problems

Reduce or eliminate data movement with in-database analytics

- **Leverage built-in extensibility mechanisms to allow secure execution of scripts**

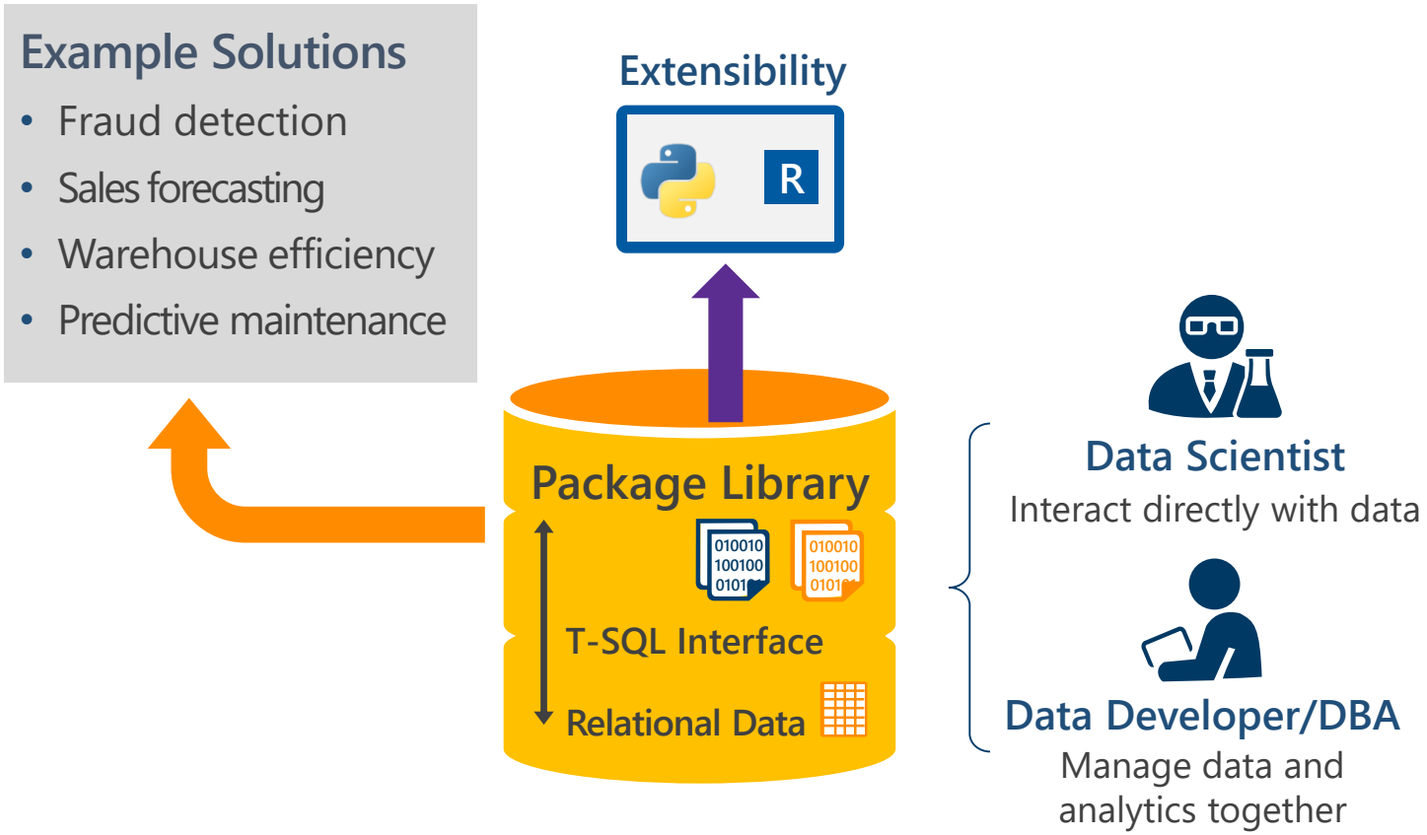
Deploy R or Python scripts and models

- **Use familiar T-SQL stored procedures to invoke scripts from your app**
- **Embed the returned predictions and plots**

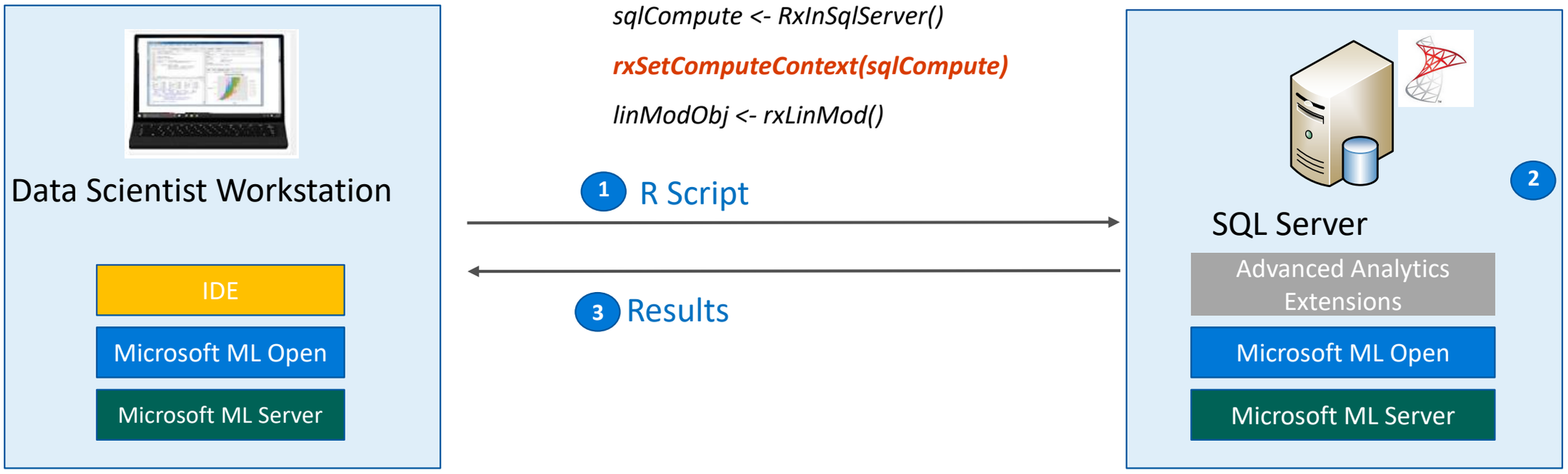
Achieve enterprise scale and performance

- **Use parallelism query capabilities of in-memory and ColumnStore indexes**
- **Leverage RevoScaleR and RevoScalePy support for large datasets and parallel algorithms**

SQL Server Machine Learning Services



Data Scientist Scenario



SQL Server Developer Scenario

Application

1 System Stored Procedure

3 Results: scores, plots

SQL Server

Advanced Analytics Extensions

Microsoft ML Open

Microsoft ML Server

2 The stored procedure contains R code and executes near-db.

```
exec sp_execute_external_script
  @ language = 'R'
, @script =
-- R code --
```

ML Server functions & algorithms

Data step

Data import – Delimited, fixed, SAS, SPSS, OBDC
Variable creation & transformation
Recode variables
Factor variables
Missing value handling
Sort, merge, split
Aggregate by category (means, sums)

Descriptive statistics

Min/max, mean, median (approx.)
Quantiles (approx.)
Standard deviation
Variance
Correlation
Covariance
Sum of squares (cross-product matrix for set variables)
Pairwise cross tabs
Risk ratio & odds ratio
Cross-tabulation of data (standard tables & long form)
Marginal summaries of cross tabulations

Statistical tests

Chi Square Test
Kendall Rank Correlation
Fisher's Exact Test
Student's t-Test

Sampling

Subsample (observations & variables)
Random sampling

Predictive models

Sum of squares (cross-product matrix for set variables)
Multiple linear regression
Generalized linear models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
Covariance & correlation matrices
Logistic regression
Classification & regression trees
Predictions/scoring for models
Residuals for all models

Simulation

Simulation (e.g., Monte Carlo)
Parallel random number generation

Cluster analysis

K-Means

Classification

Decision trees
Decision forests
Gradient-boosted decision trees
Naïve Bayes

Custom parallelization

PEMA-R API
rxDataStep
rxExec

sp_execute_external_script

```
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> [ ,...n ] ]
[;]
```

Demo

Customer Usage Drivers

Reduce data movement

Reduce complexity by writing once in R

Use RevoScale optimized function equivalents

Use compute context for exploration activities, leveraging server's resources

Leverage SQL Server design optimizations

Operationalize with sp_execute_external_script

PROS, Inc.

Revenue & Profit Realization

PROS provides a real-time software solution platform to help companies drive pricing & sales effectiveness

Reduce Quote Turnaround Time

Simplify Complex Product Catalogs

Raise Win Rates

Grow Deal Sizes

Increase Margins

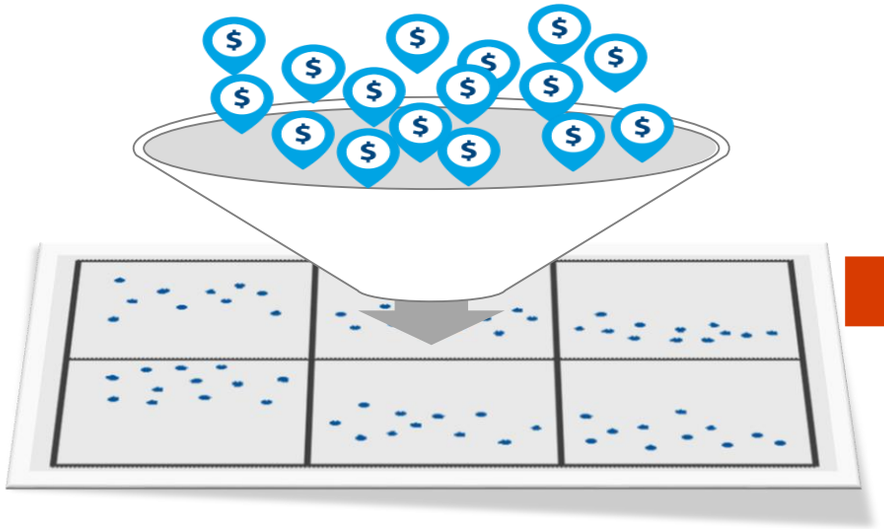
Reduce Sales Cycle Duration

Increase Quota Attainment

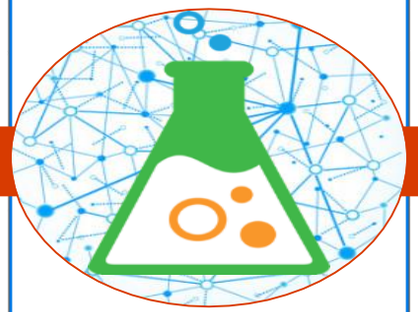
Segmentation

Foundation for PROS smart pricing guidance

Segmentation



Group customers, products & transactions into micro-segments of similar willingness-to-pay

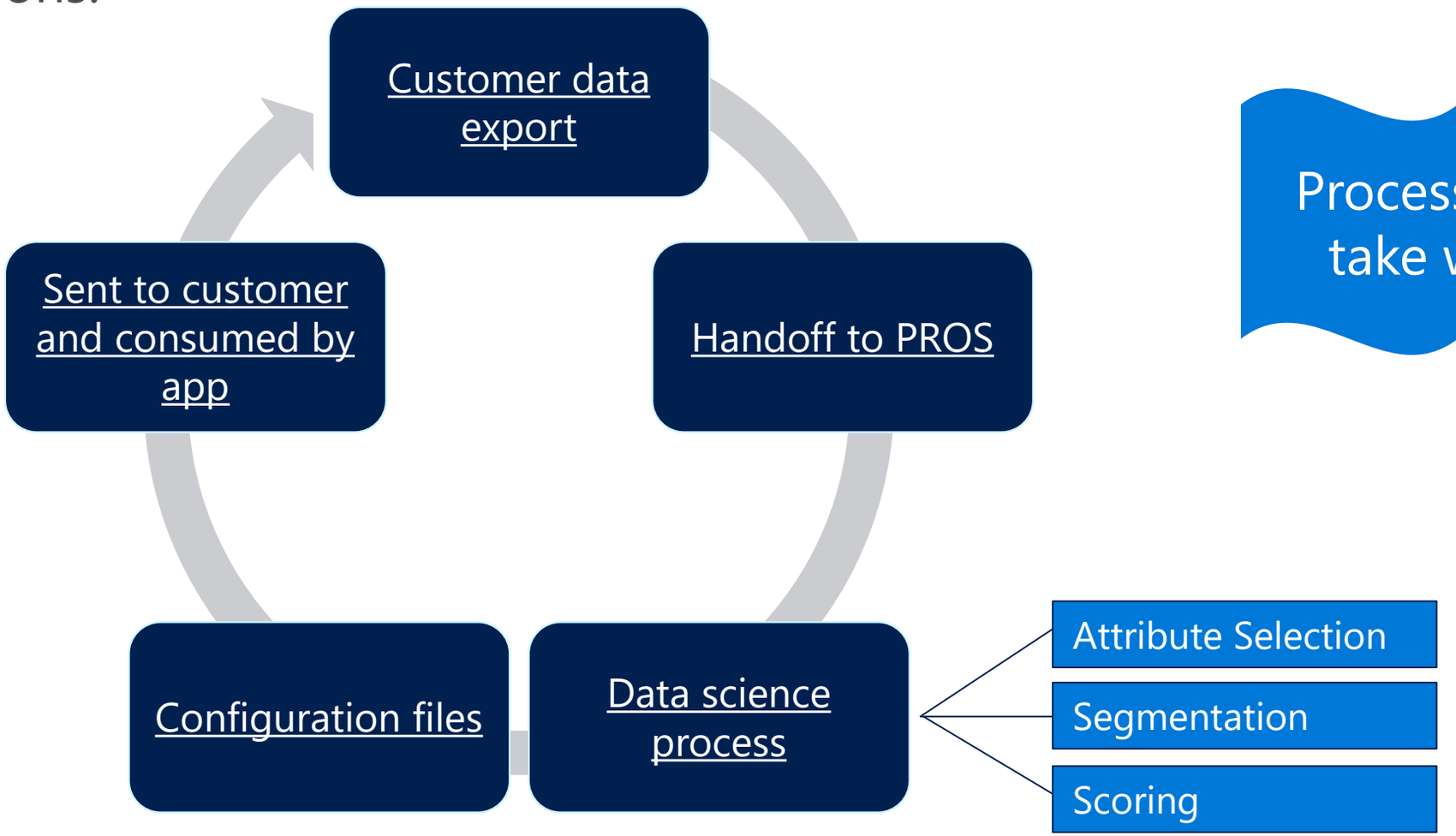


Pricing Optimization

Apply optimization algorithms to target the pricing envelope ‘sweet spot’ in every segment

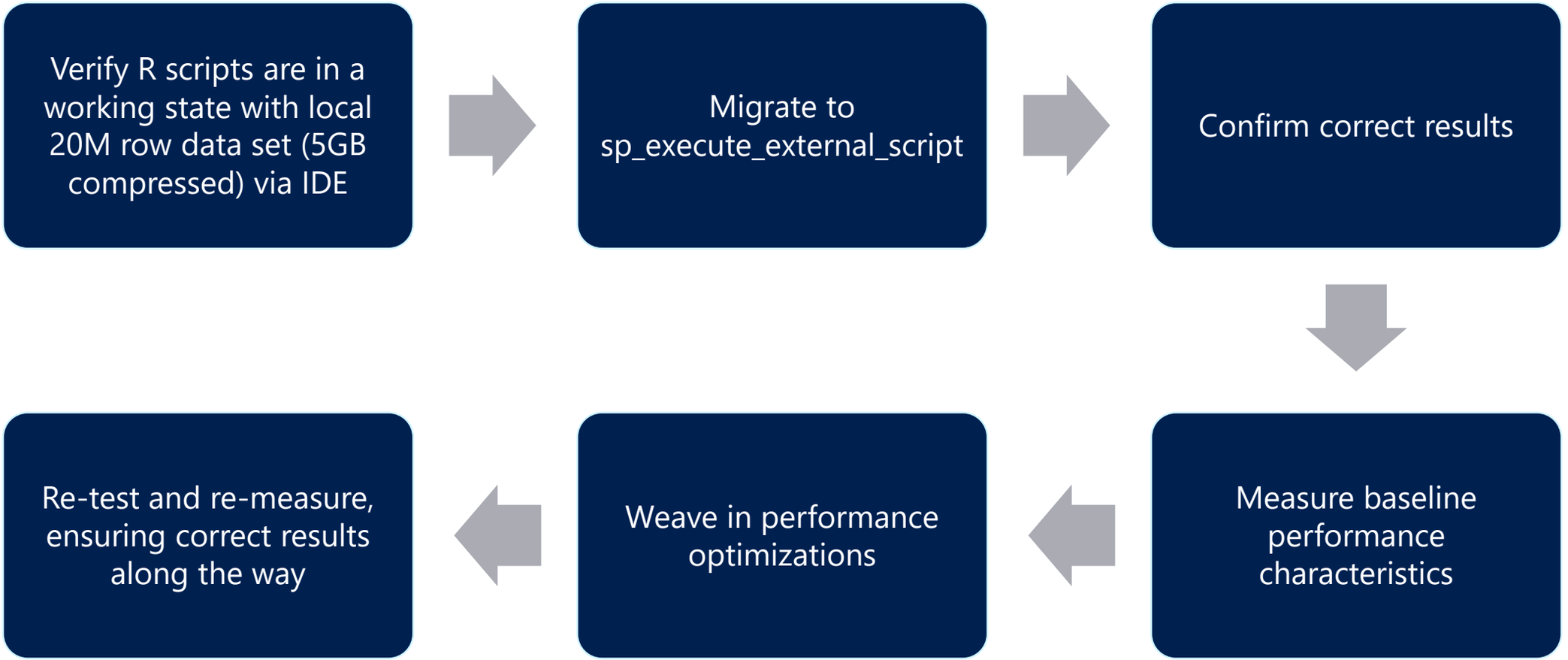
Pricing segmentation: Process

Identify appropriate set of business attributes and model how they can be used with statistical and practical soundness to provide intelligent **pricing benchmarks** for future transactions.

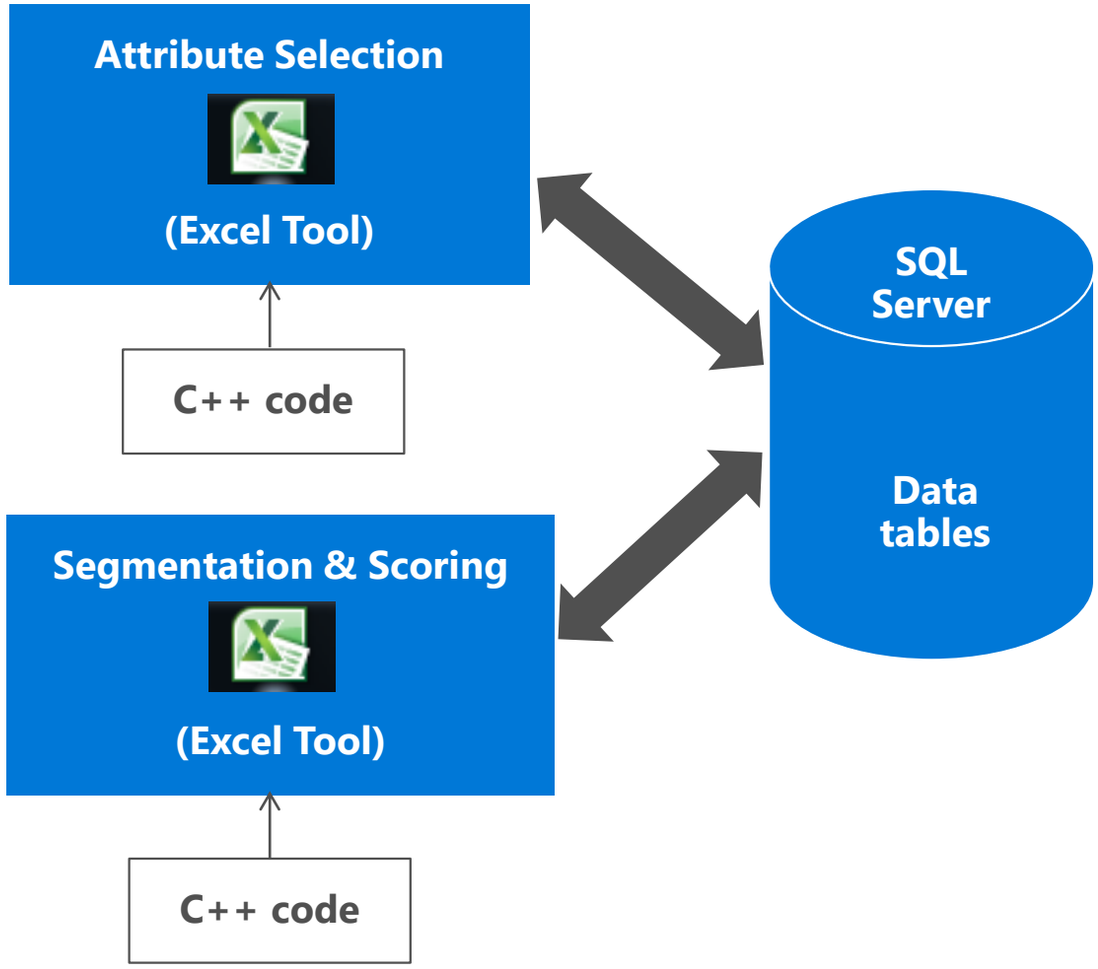


Solution

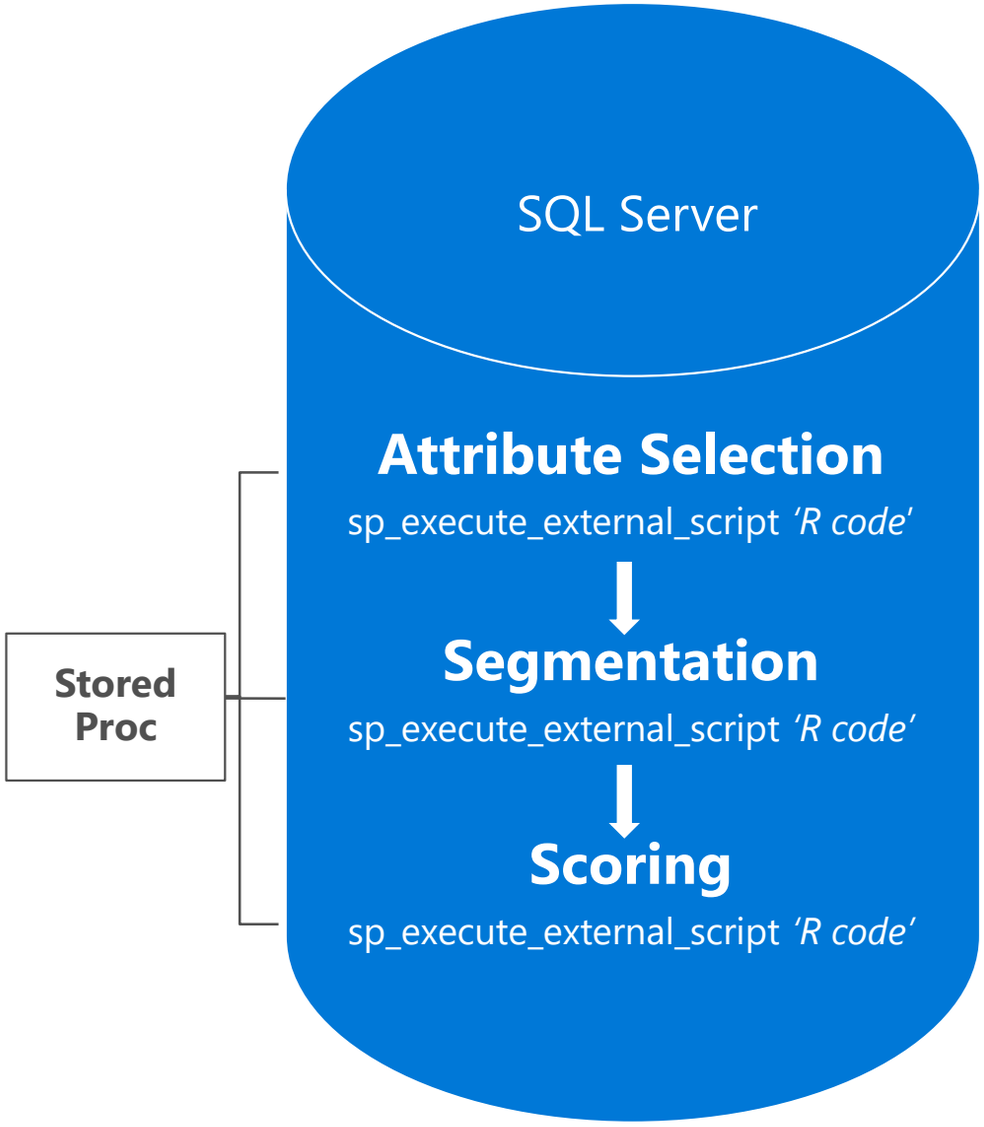
Using SQL Server R Services to improve Segmentation process performance (Excel/C++/Python tools as baseline)



Before



After



Before

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function | Addins

attribute_selection.R

Source on Save | Run | Source

```
con <- paste("Driver={SQL Server};Server=", instance_name, ";Database=", database_name, ";Trusted_Connection=true;", sep="")

rxSetComputeContext(
  RxInSqlServer(
    connectionString = con,
    autoCleanup = TRUE,
    numTasks = 8,
    consoleOutput = TRUE
  )
)

qq <- featureEngineeringQuery

attributeDataFactors <-
  RxSqlServerData(sqlQuery=qq, connectionString=con, rowsPerRead = 200000, stringsAsFactors = TRUE)

colClassesParam = rxCreateColInfo(data = attributeDataFactors, sortLevels = TRUE)

attributeData <-
  RxSqlServerData(sqlQuery=qq, connectionString=con, rowsPerRead = 200000,
                  colInfo = colClassesParam)

forest = rxDForest(featureFormula,
                   data = attributeData, nTree = 100, importance = T)

imp = forest$importance[order(forest$importance/sum(forest$importance),decreasing=T), ]/
  sum(forest$importance)

names = row.names(forest$importance)[order(forest$importance/sum(forest$importance),decreasing=T)]

OutputDataSet <- data.frame(Att = names, Imp = imp)
```

31:52 (Top Level) R Script

After

attribute_selection.sql - JSILVER-PC2.master (PROS1\jsilver (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

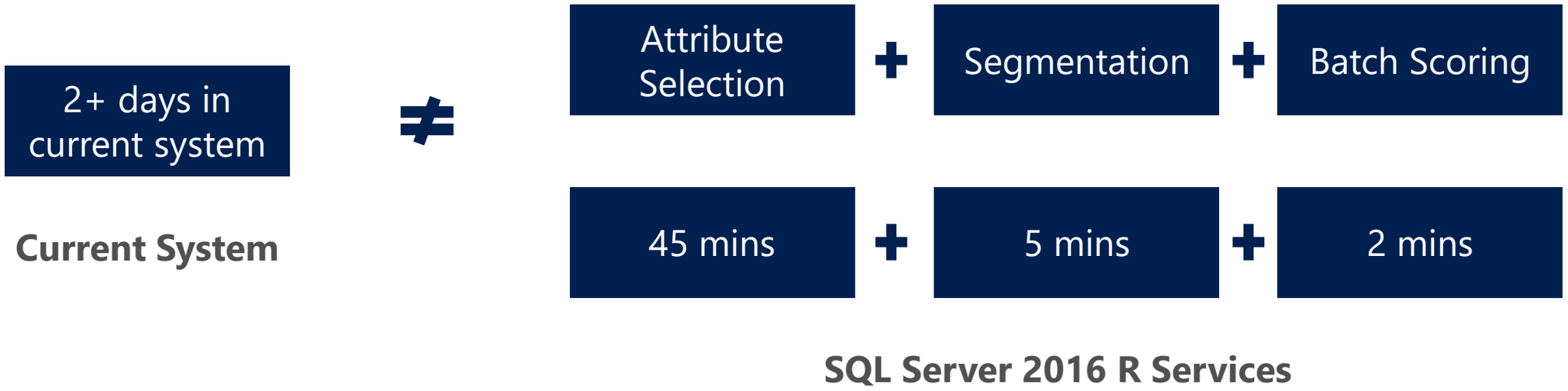
New Query | Execute | Debug | master

attribute_selection.s...(PROS1\jsilver (54))

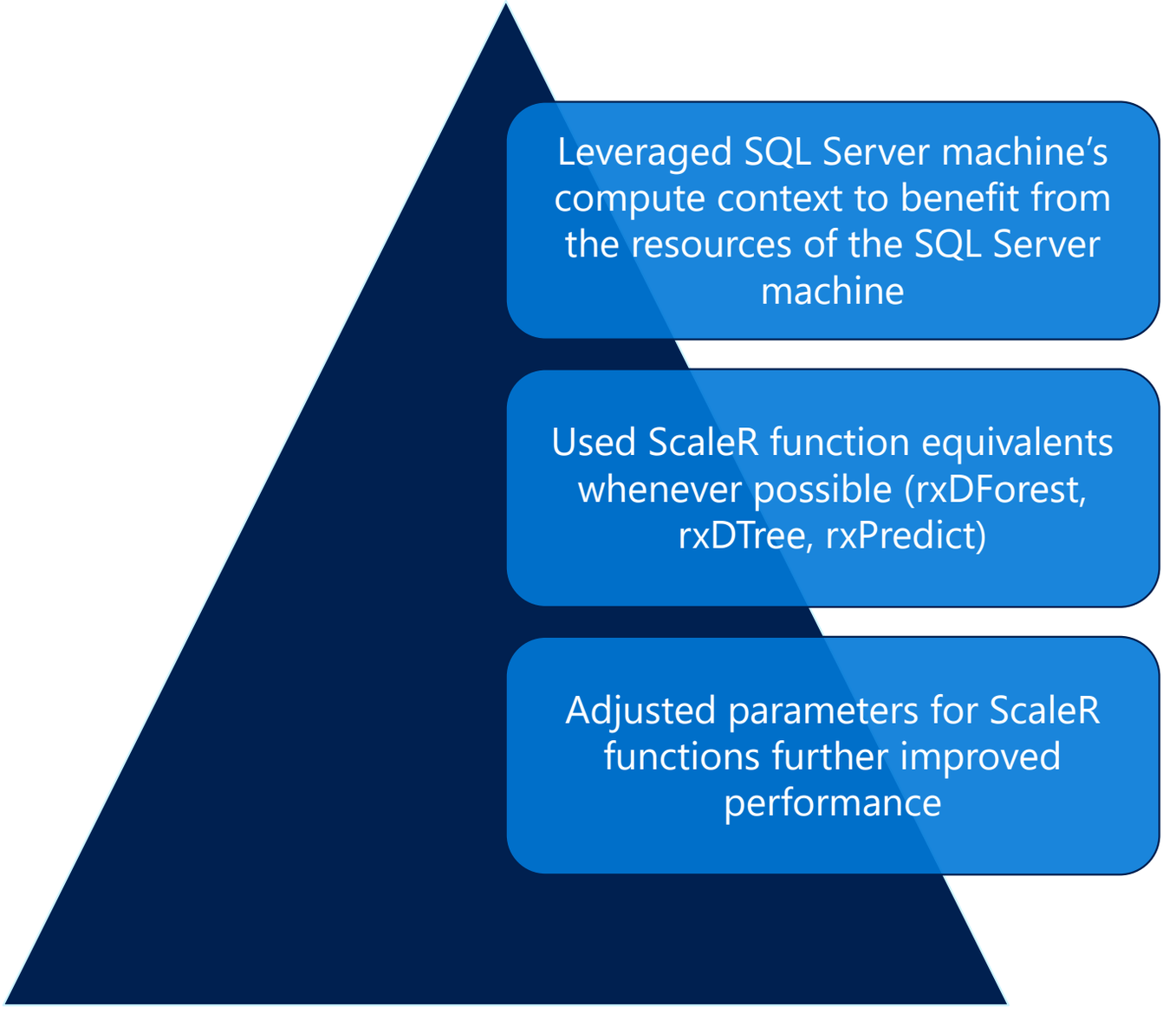
```
DECLARE @instance_name nvarchar(100) = @@SERVERNAME,
        @database_name nvarchar(128) = DB_NAME();

EXECUTE sp_execute_external_script
  @language = N'R'
, @script =
        N'
        , @input_data_1 = N''
        , @params = N'@instance_name nvarchar(100), @database_name nvarchar(128), @featureEngineeringQuery nvarchar(max), @featureFormula nvarchar(max) '
        , @instance_name = @instance_name
        , @database_name = @database_name
        , @featureEngineeringQuery = @inputQuery
        , @featureFormula = @formula
WITH RESULT SETS (([ATTRIBUTE] varchar(256), [IMPORTANCE] float NULL));
GO
```

100 %

The results


2+ days in current system
≠
Attribute Selection
+
Segmentation
+
Batch Scoring
45 mins
+
5 mins
+
2 mins
Current System
SQL Server 2016 R Services

How was this achieved?



ATTOM Data Solutions & Greenfield Advisors

Company overview

- ATTOM Data Solutions is a leading provider of property data - including tax, deed, mortgage, foreclosure, environmental risk, natural hazard, health hazard, neighborhood characteristics and property characteristics – for more than 150 million U.S. properties.

- Greenfield Advisors is a real estate and business consulting firm headquartered in Seattle, Washington. They are internationally recognized in the real estate appraisal profession as the leading authorities on the analysis and valuation of property impacted by environmental factors.

Property valuation

The property valuation process needs to provide timely and accurate calculations

- **Over 100 million subject residential properties**

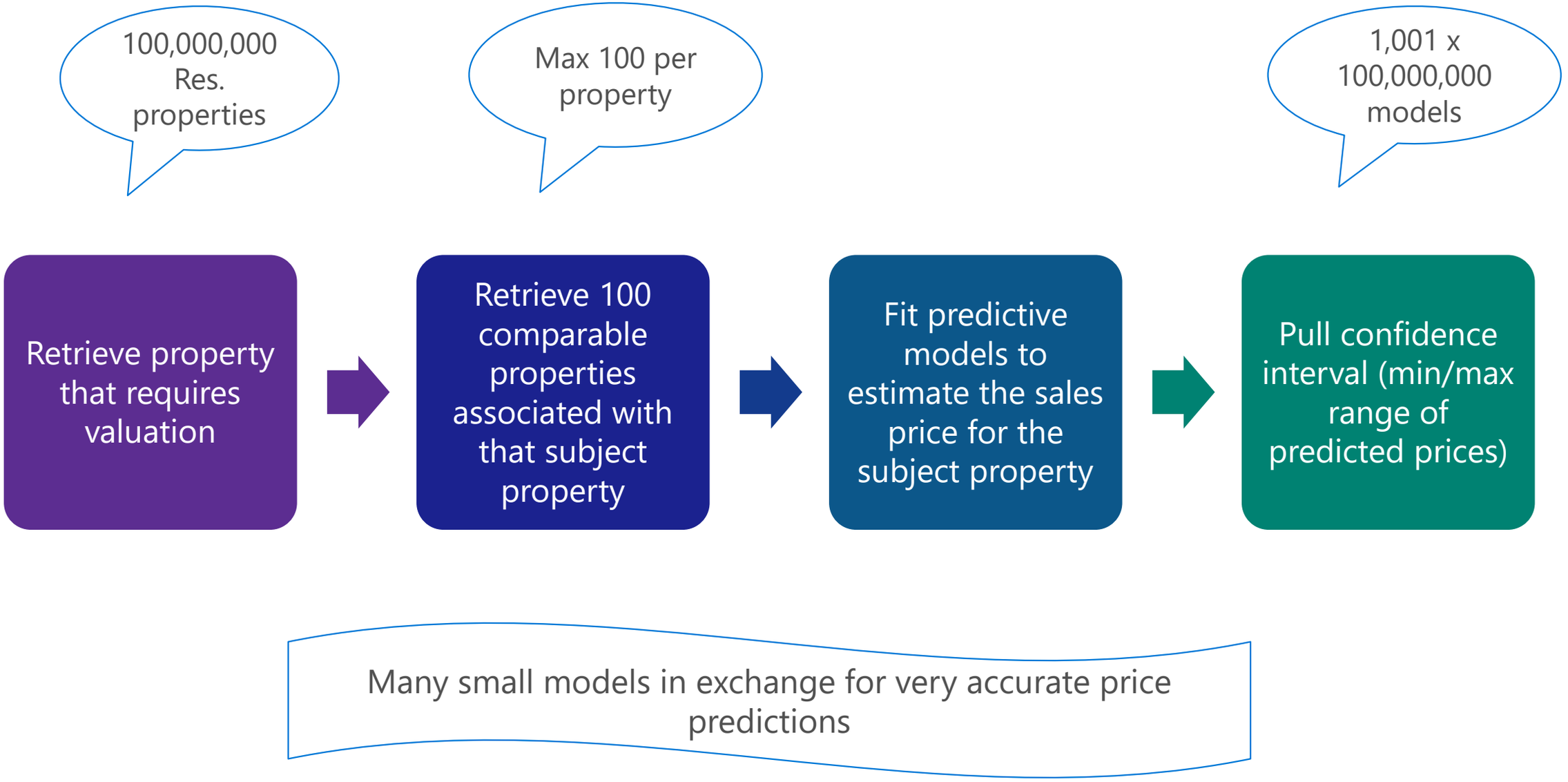
Initial implementation: limited to ~ 200 valuations per hour

- **7 R script files, all Open R, ~ 600 lines of code**

Objectives

- Port the existing legacy code to SQL Server R Services
- Gain deep understanding of existing code
- Look for refactoring opportunities (R and SQL)
- Test maximum throughput capabilities

Data flow


100,000,000 Res. properties
Max 100 per property
1,001 x 100,000,000 models
Retrieve property that requires valuation
Retrieve 100 comparable properties associated with that subject property
Fit predictive models to estimate the sales price for the subject property
Pull confidence interval (min/max range of predicted prices)
Many small models in exchange for very accurate price predictions

The results

- How was this achieved?
  - Putting data and computation in same place was automatic win
  - Pre-ca[illegible]100 compar[illegible] T-SQL and a cache
    - At th[illegible]d in a jump [illegible]ur to 660k valu[illegible]
  - Conc[illegible]ecute_exte[illegible]ts
- DBA a[illegible]expedited a[illegible]
  - Doing [illegible]ted exercise [illegible] allowed for progress in a short time window

Before: limited to ~ 200 valuations per hour

After: Using SQL Server R Services, able to scale to ~ 720,000 valuations per hour

What about DBA use cases?

Predicting capacity over the holiday

Identifying outliers in error logs

Characterizing workloads

Analysis of upgrade-regression testing

Query Store data mining

Rich visualizations

Microsoft Advanced Analytics Landscape

Azure Machine Learning

- Fully managed cloud service that enables you to easily build, deploy, and share predictive analytics solutions

SQL Server Machine Learning Services (In-Database)

- Supports both R and Python pushed SQL Server compute-context

Microsoft R Server

- For enterprise-level R deployments on Windows and Linux servers

Microsoft Machine Learning Server

- Supports R and Python deployments on Windows servers, with expansion to other supported platforms planned for late 2017

Thank you