FROM THE ORGANISERS OF **SOFTWARE ARCHITECT**

DEVWEEK2015

Monday 23 – Friday 27 March | Central Hall, Westminster, London

Introducing R and Azure ML

Dejan Sarka

Instructor Bio

Dejan Sarka (dsarka@solidq.com)

- 30 years of experience
- SQL Server MVP, MCT,…
- 13 books
- 7+ courses
- Focus:
  - Data modeling
  - Data mining
  - Data quality

Agenda

- Introducing R
- Introducing Azure ML

About R (1)

- The R statistical programming language is a free open source package based on the S language developed by Bell Labs
- R written as a research project by Ross Ihaka and Robert Gentleman
  - Now developed by a group of statisticians called 'the R core team', with a home page at www.r-project.org
- R is available free of charge and is distributed under the terms of the Free Software Foundation's GNU General Public License
  - Available for Windows, Mac OS X, and Linux

About R (2)

- R can run interactively
- R is a programming language for analyzing data
  - Many statistical functions are already built in
  - Excellent graphic functionality
  - Contributed packages expand the functionality to cutting edge research
- Some drawbacks as well
  - Since it is a programming language, generating computer code to complete tasks is required
  - Used to be the sole province of academic statisticians
  - It's open – different procedures for the same task

Getting R

- Install R from r-project.org
  - UAC not being triggered
  - R packages extend the language - you need to be able to download zip files
  - Regular updates
- R defaults to an interactive mode
- R Console – command prompt or GUI?
  - A prompt “>” is presented to users
  - Each input expression is evaluated and a result returned

Introducing RStudio

- RStudio IDE is a powerful and productive user interface for R
  - It's free and open source, and works on Windows, Mac, and Linux
- Install it from Rstudio.com
  - Installation does not trigger UAC
  - Regular updates

R Language Basics (1)

- R is a *functional* language
- You don't type commands but rather call *functions* to achieve results, even quit

  > q()
- Other common functions

  > help(<topic>) or ?<topic>

  > license()

  > contributors() and citation()

  > options() e.g. options(cmdhelp=TRUE) to get compiled help (default installation option)

  > source() code from file and sink() results to a file

R Language Basics (2)

- R is *case sensitive*
- Comments can be put almost anywhere, starting with a hash mark (‘#’)
- Commands are separated either by a semi-colon (‘;’), or by a newline
  - Commands can be grouped together into one compound expression by braces (‘{’ and ‘}’)
- The entities that R creates are known as *objects*
  - The collection of current objects is the *workspace*

  > objects() to list the current objects

  > rm(<object>) to remove an object from the workspace

Storing Code and Objects

- At the end of each R session you are given the opportunity to save all the currently available objects
  - The objects are written to a file called .RData in the current directory, and the command lines used in the session are saved to a file called .Rhistory
- RStudio can work with script files
  - Called .R

R Expressions and Variables

- Basic expressions
  - > 1 + 1
  - > 2 + 3 * 4
  - > 3 ^ 3
  - > sqrt(81)
  - > pi
- Variables
  - Numeric, Boolean, Strings
  - Type determined automatically
  - Created with "<-" operator
  - Name is case sensitive and can include a period

R Vectors

- Vectors are ordered collections of numbers
- Created with
  - > c() to concatenate elements or sub-vectors
  - > rep() to repeat elements or patterns
  - > seq() or m:n to generate sequences
- Most mathematical functions and operators can be applied to vectors without loops
- Use the [] operator to select elements
  - Select or exclude specific elements

Other Collections and Objects

- *Matrices* or more generally *arrays* are multi-dimensional generalizations of vectors
- *Factors* provide compact ways to handle categorical data – distinct values are *levels*
- *Lists* are a general form of vector in which the various elements or *components* need not be of the same type
- *Data frames* are matrix-like structures, in which the columns can be of different types
- *Functions* can be stored in the project's workspace - a simple way to extend R

Using SQL Server Data in R

- Get a SQL Server ODBC drive
- Create a system DSN
- Install RODBC package for R
- Activate the package
- Create a connection object
- Read the data into a data frame

```
con <- odbcConnect("AWDW2014", uid="RUser",
   pwd="Pa$$w0rd")
df_TM <- as.data.frame(sqlQuery(con,
  "SELECT CustomerKey, MaritalStatus, Gender,
    Region, BikeBuyer
   FROM dbo.vTargetMail"), stringsAsFactors = FALSE)
```

data.table

- Additional package
- The data.table object inherits from data.frame
- SQL-like and fast
  - Column names
  - Key (allows duplicates, defines sort order)
  - Aggregations
  - Grouping
  - Filtering
  - Joins – fast ordered joins aka last observation carried forward (LOCF) joins aka rolling joins – merge joins

Basic Graphics

- Histogram with a title and axis labels and color

```
hist(dt_TM[,NumberCarsOwned], main = 'Number of Cars Owned',
     xlab = 'Number of Cars Owned', ylab = 'Number of Cases',
     col="purple")
```

- Plot with two lines, title, legend, and axis legends

```
plot(dt_TM[(CustomerKey < 11010),NumberCarsOwned],
     type="o",col='blue', xlab="Key", ylab="Number")
lines(dt_TM[(CustomerKey < 11010),TotalChildren],
      type="o",col='red')
legend("topleft",
       names(dt_TM[, list(NumberCarsOwned,TotalChildren)]),
       cex=0.8,col=plot_colors,
       lty=1:2,lwd=1, bty="n");
title(main="Two Variables Line Chart",
      col.main="DarkGreen", font.main=4)
```

Basic Statistics

- Summary of the dataset

```
summary(dt_TM)
```

- Some centers

```
sapply(list(dt_TM[,NumberCarsOwned],
 dt_TM[,TotalChildren]),mean)
sapply(list(dt_TM[,NumberCarsOwned],
 dt_TM[,TotalChildren]),median)
```

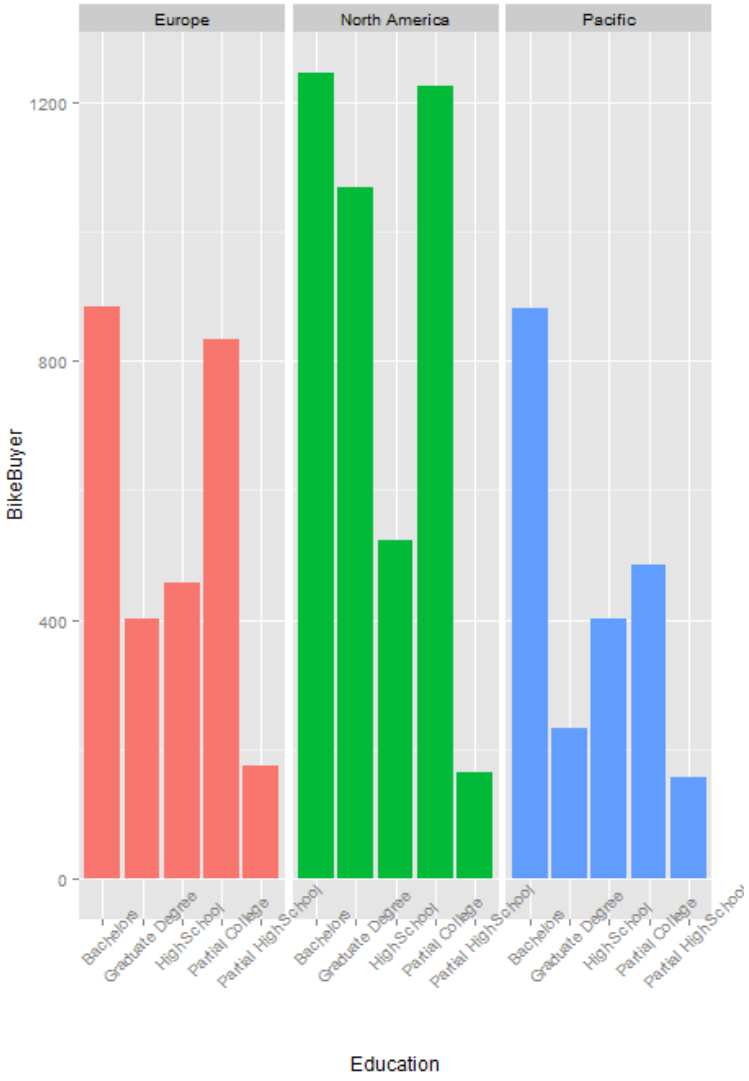
- Average and count with group by

```
aggregate(dt_TM[,NumberCarsOwned],
          by=list(dt_TM[,CommuteDistance]),
          FUN=mean)
```

ggplot2

- ggplot2 implements a grammar of graphics (gg)
- Simple, consistent functions to build charts

```
ggplot(data=dt_TM,
aes(x=Education,y=BikeBuyer,
fill=Region))
geom_bar(stat="identity")
facet_grid(.~Region)
theme(legend.position="none",
axis.text.x=
element_text(angle=45))
```



Data Mining in R

- Many, many algorithms in different packages
  - All popular algorithms
  - Can become confusing

```
# Package party (Decision Trees)
install.packages("party", dependencies = TRUE)
library("party")
# Train the model
TMDT <- ctree(BikeBuyer ~ NumberCarsOwned + Region,
              data = df_TM)
# Show the results
plot(TMDT, type = "simple")
```

Why Machine Learning

- Satya Nadella: “I believe over the next decade computing will become even more ubiquitous and intelligence will become ambient...This will be made possible by an ever-growing network of connected devices, incredible computing capacity from the cloud, insights from big data, and intelligence from machine learning”
- Bill Gates: “If you invent a breakthrough in Artificial Intelligence, so machines can learn, that is worth 10 Microsofts”

What Is Machine Learning?

- Formal definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E” - Tom M. Mitchell
- Another definition: “The goal of machine learning is to program computers to use example data or past experience to solve a given problem.” – Introduction to Machine Learning, 2nd Edition, MIT Press

Classes of ML Problems

- Classification: Assign a category to each item
- Regression: Predict a real value for each item
- Ranking: Order items according to some criterion
- Clustering: Partition items into homogeneous groups
- Dimensionality reduction: Transform an initial representation of items into a lower-dimensional representation while preserving some properties

Machine Learning vs Data Mining

- Tom Dietterich: “The goal of machine learning is to build computer systems that can adapt and learn from their experience”
- Michael J. A. Berry and Gordon S. Linoff: “Data mining is the process of exploration and analysis, by automatic or semiautomatic means, of large quantities of data in order to discover patterns and rules”
- ML machine oriented, DM people oriented?
- But…

Machine Learning vs Data Mining

- ML primary techniques:
  - Supervised (or directed) learning
  - Unsupervised (or undirected) learning
- DM primary techniques:
  - Directed (or supervised) learning
  - Undirected (or unsupervised) learning
- Great things happen in machine learning when human and machine work together
  - Combining a person's knowledge of relevant features with the machine's talent for optimization
- ML and DM are nearly synonyms

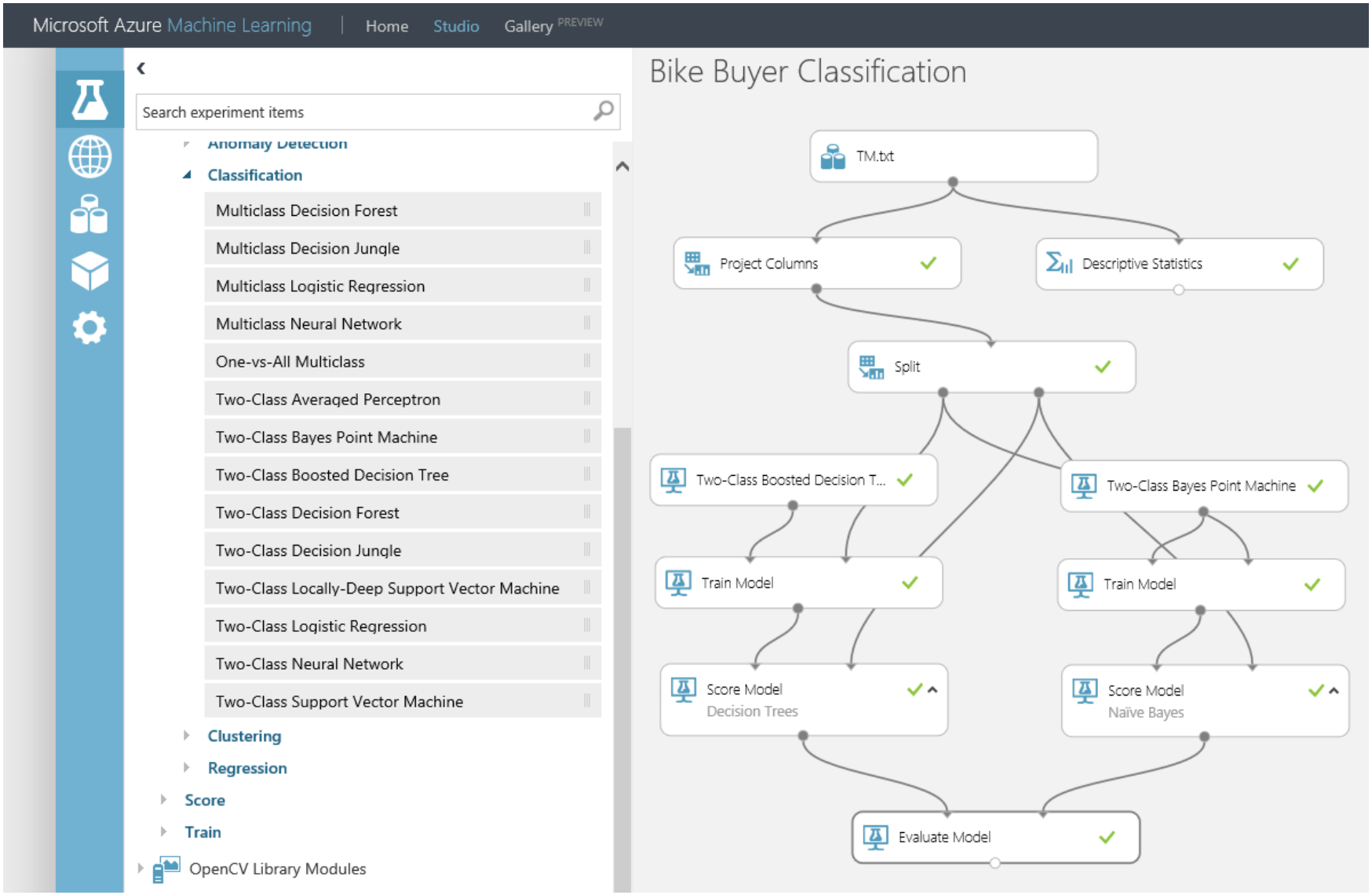
Introducing Azure ML

- Accessible through a web browser, no software to install
- Collaborative work with anyone, anywhere via Azure workspace
- Visual composition with end to end support for data science workflow
- Built-in ML algorithms
- Extensible, support for R

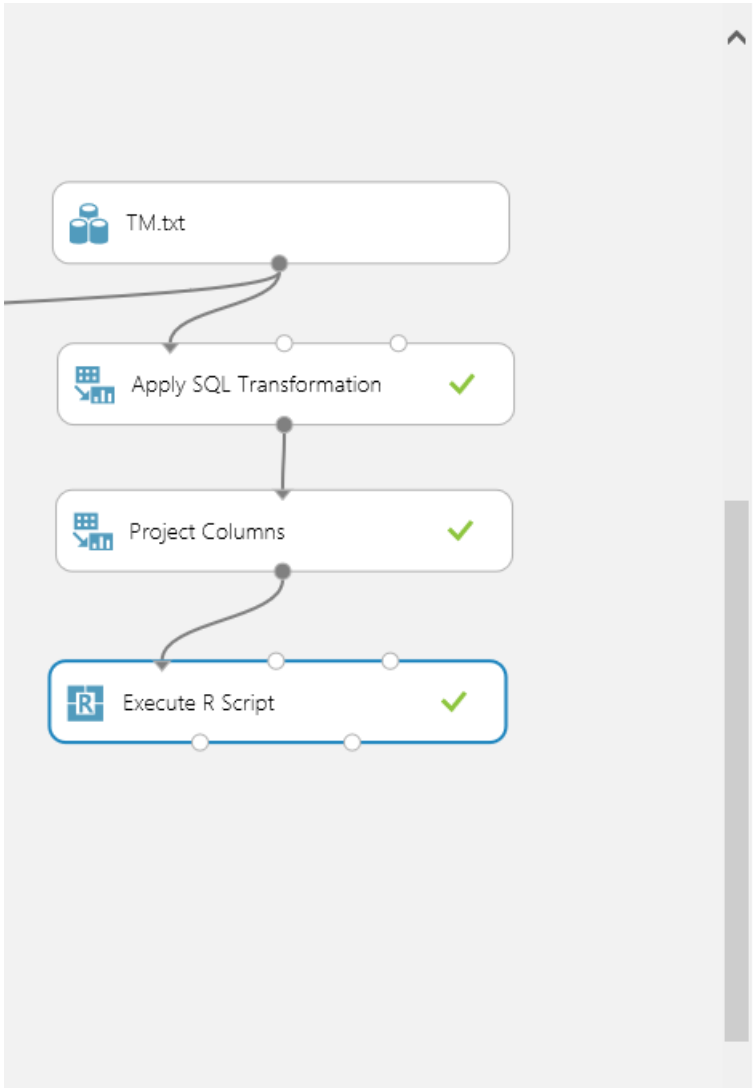
Azure ML Workflow and Components

- Workflow
  - Upload, import online, or connect to some current or historical data
  - Build and validate a model
  - Create a web service that uses your trained models to make live predictions
- Components
  - Experiments
  - Web services
  - ML Studio

Azure ML Experiment



Extensibility



Execute R Script

R Script

```
# load("src/yourData.rdata");

# create a distance matrix from the data
ds <- dist(df_TM, method = "euclidean")

# Hierarchical clustering model
TMCL <- hclust(ds, method="ward.D2")

# Display the dendrogram
plot(TMCL)

# Cut tree into 6 clusters
groups <- cutree(TMCL, k=6)

# Draw dendogram with red borders around the 6 clusters
rect.hclust(TMCL, k=6, border="red")


# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("df_TM");
```

Random Seed

START TIME	3/21/2015 4:34:48 PM
END TIME	3/21/2015 4:34:48 PM
ELAPSED TIME	0:00:00.000
STATUS CODE	Finished
STATUS DETAILS	Task output was present in output cache

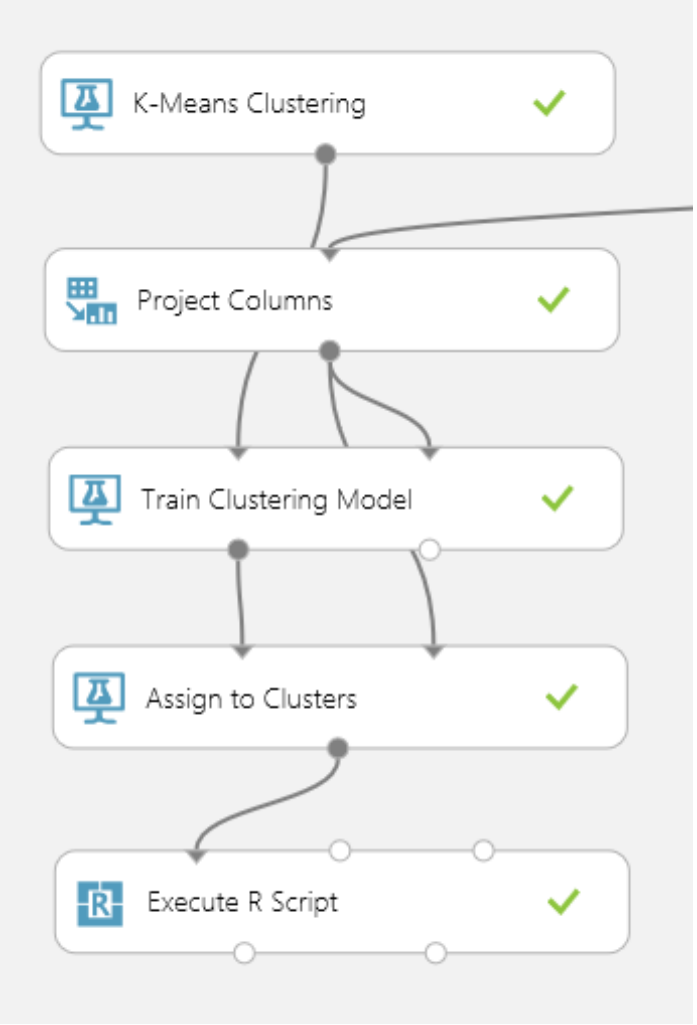
Azure ML Pros

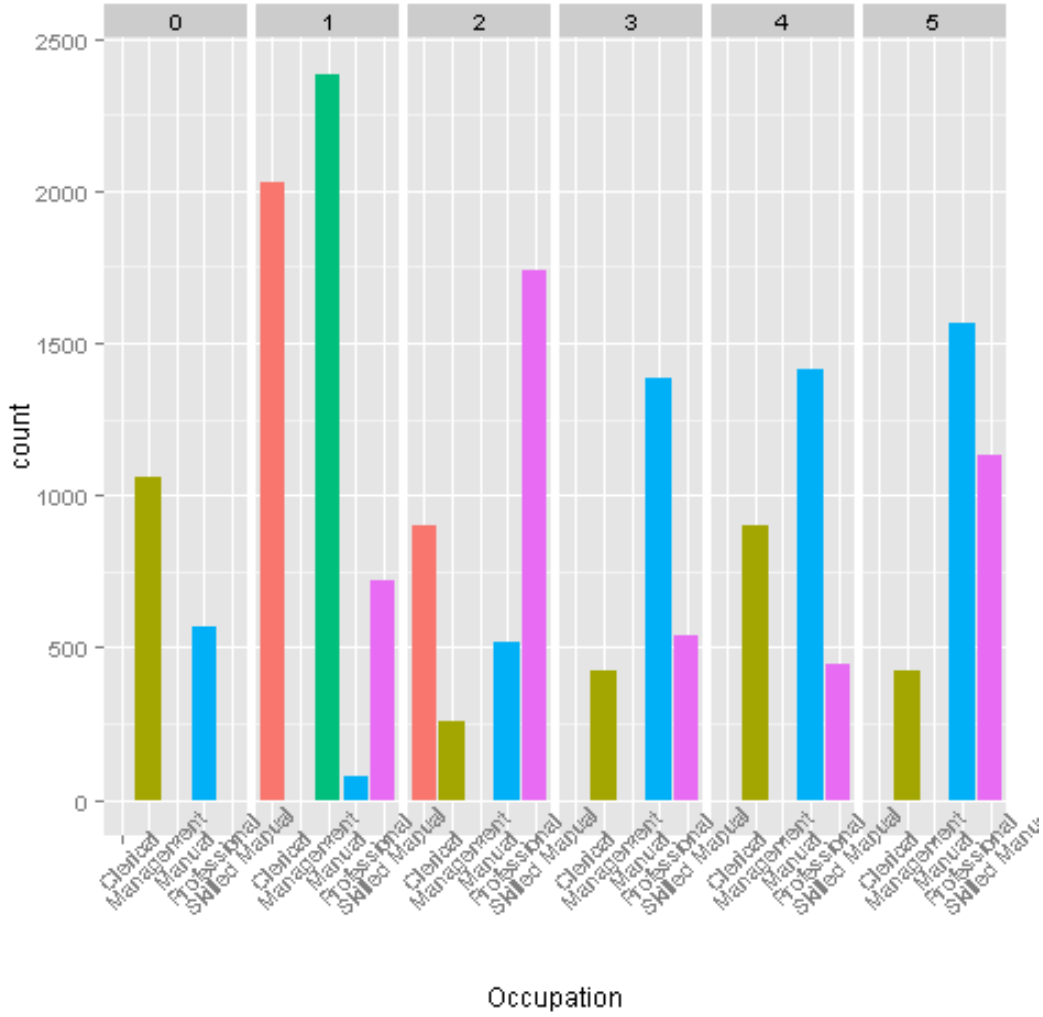
- Reduce complexity to broaden participation
- Immutable library of models, search discover and reuse
- Rapidly try a range of features, ML algorithms and modeling strategies
- Quickly deploy model as Azure web service to ML API service

Azure ML Cons

- No models visualizations
  - Can use R
- Cost: Per-hour fee is lower whilst you are using ML Studio ($0.38/hour) and a little higher when in production via ML API Service ($0.75/hour). The per-API calls are free while in ML Studio and cost $0.18/1000 predictions while in production
  - Might get costly if you are doing online predictions for a busy system, e.g. for fraud detection

Azure ML and R Visualizations





Questions?

Thank you!