

Lab Guide

AI with IBM RPA

Nigel T. Crowther

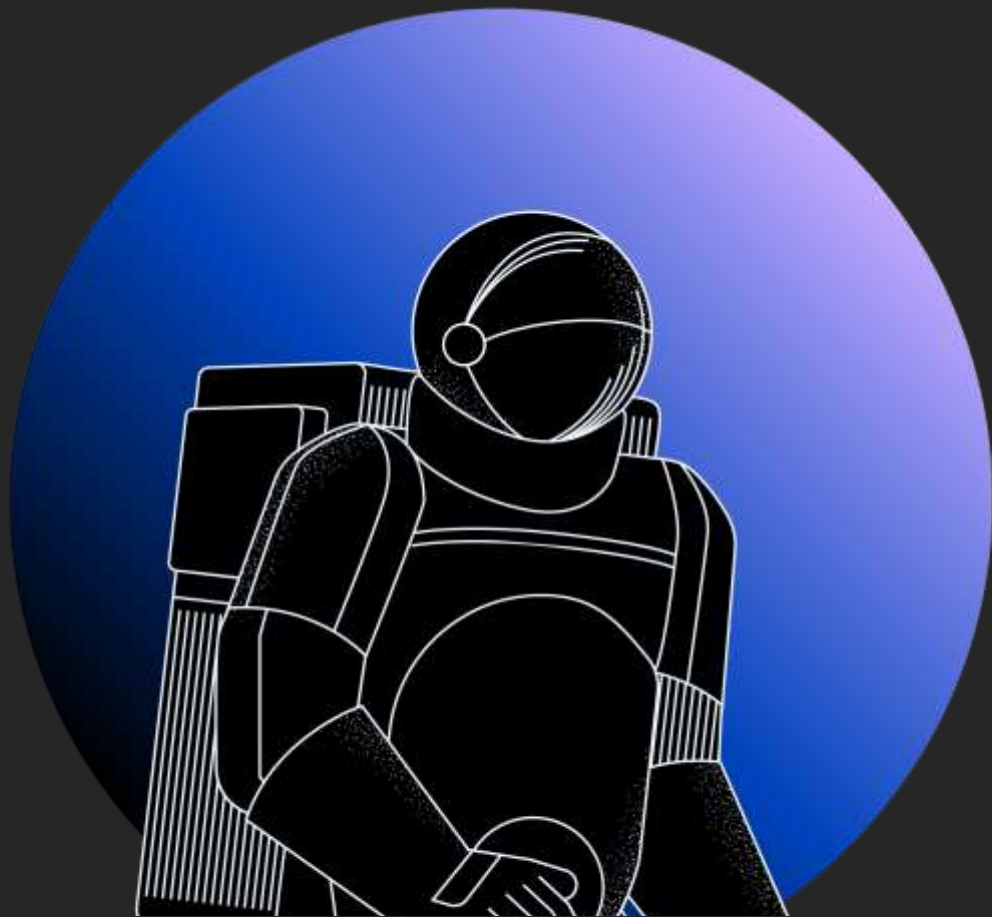
ncrowther@uk.ibm.com

Jukka Juselius

jukka.juselius@fi.ibm.com

Hands-on Lab

Version 1.0 for General Availability





NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



A little learning is a dangerous thing;
Drink deep, or taste not the Pierian spring
Alexander Pope



Table of Contents

1 Introduction.....	6
1.1 Use Case.....	6
1.2 Prerequisites	6
2 Getting Started	7
2.1 Log In.....	7
3 First Scenario – Fuzzy Comparison	8
3.1 Scenario Description	8
3.2 Open a work in progress script	8
3.3 Company Addresses Spreadsheet	9
3.4 Adding Commands.....	9
3.5 Run the Script.....	11
4 Second Scenario Part 1 – Knowledge Base Creation	13
4.1 Scenario Description	13
4.2 AI - Real World Decisions	13
4.3 Build Knowledge Base	13
4.4 Upload Knowledge Base	14
4.5 Build Script.....	15
4.5.1 Input Box Command	15
4.5.2 Answer Question Command.....	16
4.5.3 Show Message Box Command	17
4.6 Run Script.....	18
5 Second Scenario Part 2 – Knowledge Base Training.....	20
5.1 Real World Alignment	20
5.2 Navigate to Knowledge Base Training.....	20
5.3 Open the Knowledge Base.....	20
5.4 Training the Knowledge Base	20
5.4.1 Add new classification.....	20
5.4.2 Test new classification	24
5.4.3 Optional extras	25
5.4.4 Reclassification	25
5.4.5 Test reclassification	26
5.4.6 Save changes.....	27
6 Third Scenario - Text Classifier.....	28
6.1 Scenario Description	28
6.2 Inspect File	28
6.3 Open a work in progress script	28
6.4 Train the Text Classifier	28
6.5 Add the Text Classifier.....	30
6.6 Run the Script.....	31
7 Fourth Scenario – Basic R Script.....	33
7.1 Scenario Description	33
7.2 Open a work in progress script	33



7.3 Run the R Script	33
7.4 If you have time	34
8 Fifth Scenario – Machine learning with R.....	35
8.1 Scenario Description	35
8.2 Install latest version of R	35
8.3 Create the predictive model	35
8.4 Open a work in progress script	38
8.5 Run the R Script	39
9 Sixth scenario – Business Rules	41
9.1 Scenario Description	41
9.2 Open a work in progress script	41
9.3 Test bot	43



1 Introduction

In this lab we will demonstrate how AI solves a wide variety of automation problems and show how easy it is to infuse AI into your bots

1.1 Use Case

Validating customer addresses

1.2 Prerequisites

None. You have everything you need in your lab environment. Let's get started!



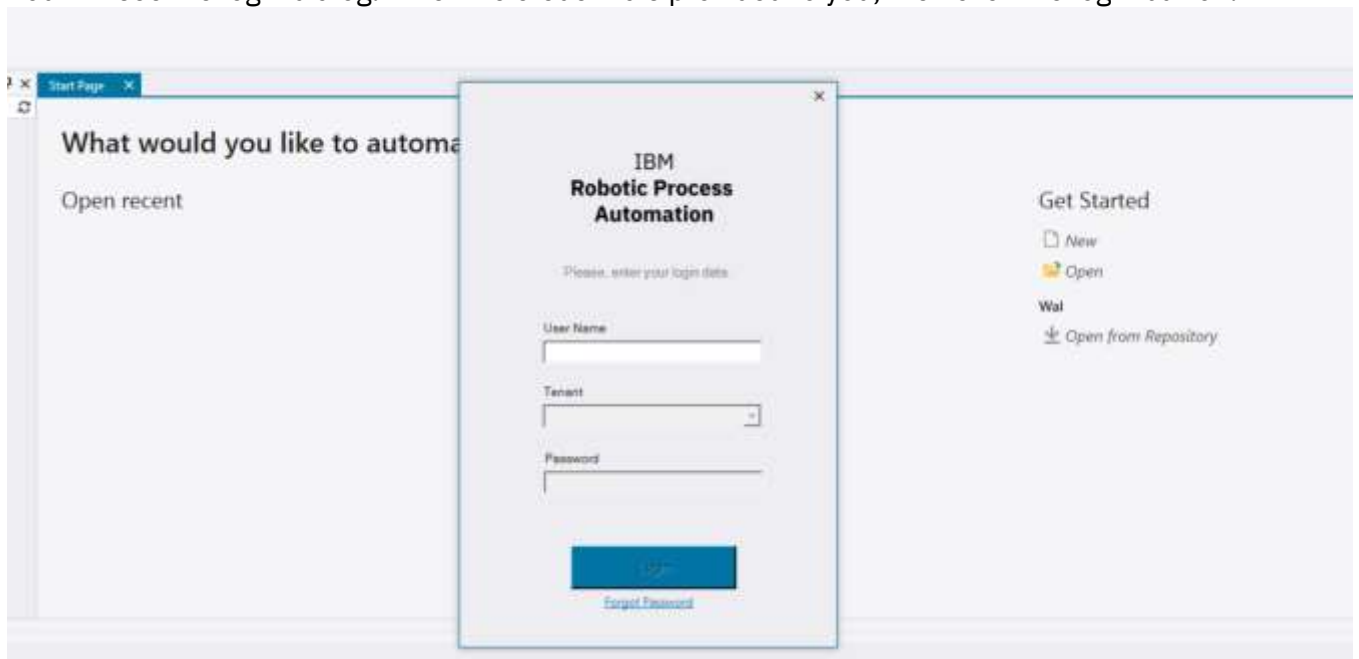
2 Getting Started

On your desktop find the IBM RPA Studio icon and launch it.



2.1 Log In

You will see the login dialog. Enter the credentials provided to you, then click the login button.



Once the tenant is retrieved, enter your password. Click login again to finish logging into the client.



3 First Scenario – Fuzzy Comparison

Fuzzy comparison uses smart algorithms to determine the similarity between strings. Similarity is graded and the higher the grade the greater the confidence. IBM RPA offers a choice of algorithms to suit different use cases.

3.1 Scenario Description

Focus Corp is a company with a tele-sales team who ask customers for an address to receive follow-up literature. The problem is that addresses are dictated and are often transcribed incorrectly, causing delivery failures.

You are an automation developer responsible for building a script to cross check addresses captured by tele-sales against known addresses. Your script should find the closest match.

3.2 Open a work in progress script

Open the folder **Lab 2 - AI with RPA\Scenario1**. Open IBM RPA Studio and select *Open* in the top left. Open *Scenario 1 Started.wal*. Your script will open in Studio.



3.3 Company Addresses Spreadsheet

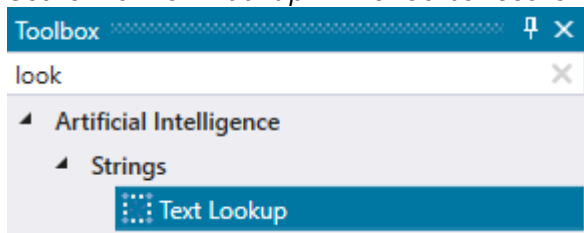
Using Libre Office, Open *CompanyAddresses.xls* situated in the same *Scenario1* folder. You should see a list of company addresses:

1	<u>CompanyAddress</u>
2	<u>Tangential Ltd,</u> <u>2 Triton Square,</u> <u>Regent's Place,</u> <u>London</u> <u>NW1 3AN</u>
3	<u>Stairs Design Ltd,</u> <u>35 Watling Street Road,</u> <u>London,</u> <u>SW1 8EA</u>
4	<u>Mealy Ltd</u> <u>164 Kings Road,</u> <u>Big In Hill,</u> <u>Westham,</u> <u>Kent</u> <u>TN16 3NH</u>

Close the spreadsheet. In RPA Studio, navigate to subroutine *matchInputAddressAgainstCandidates*. On line 20 edit the *ExcelOpen* command and make sure the command opens *CompanyAddress.xls* in the above folder.

3.4 Adding Commands

Search for *Text Lookup* in the toolbar search.



Navigate to subroutine *FindNearestMatch*. Drag the *Text Lookup* command to line 39 and fill with the information below.



Text Lookup

comment here

Input Parameters

Language ? en-US

Text* ? \${vDataEntryCompany}

Data Source* ? Data Table

Algorithm* ? Soundex

Soundex Algorithm* ? Metaphone

Data Table* ? \${vListedCustomers}

Use Column Name ? ☒

Name* ? CompanyAddress

Output

Text Found ? vMatchedCompany

Index ? vListedIndex

Success ? success

Cancel Save

Note: You should set *Data Source* to *Data Table*. You should slide *Use Column Name* to the ON position. The column name value should be *CompanyAddress* which Is not a variable.

When done, press *Save*

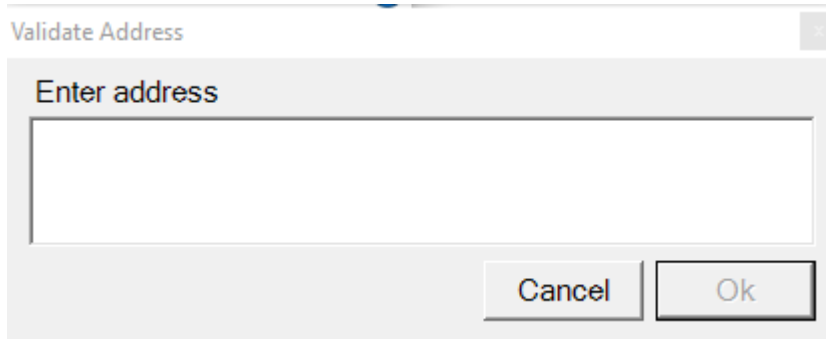
Click on the Call Graph tab. It should look like this





3.5 Run the Script

Run the script by pressing Ctrl+F5 (run without debugging).
You should see a dialog box appear.

A dialog box titled "Validate Address" with a close button in the top right corner. Inside the dialog, there is a label "Enter address" above a large text input field. At the bottom right of the dialog are two buttons: "Cancel" and "Ok".

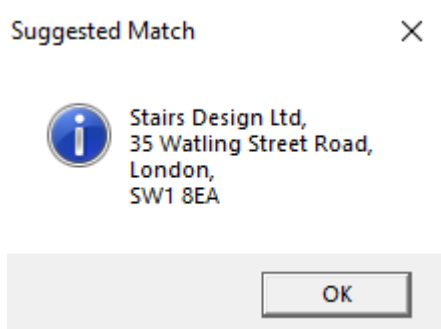
If you get an error its likely you still have the Company Addresses spreadsheet open. Close the spreadsheet and try again

Now paste in the following address:

Stairs Design Ltd,
*35 **Waddling** Street*
London,
FW1 8EF

Press OK (or ENTER).

The script performs a fuzzy match between the entered address and the database addresses. It finds a match and displays:

A dialog box titled "Suggested Match" with a close button in the top right corner. On the left is a blue circular information icon. To its right is the text: "Stairs Design Ltd, 35 Watling Street Road, London, SW1 8EA". At the bottom right is an "OK" button.

If the fuzzy match falls below the tolerance level it does not display a match.
Try matching different addresses by mistyping the addresses in the *CompanyAddress.xlsx* spreadsheet.

After testing, click on Cancel to stop the script.

IBM RPA provides several choices of algorithms based on your needs. Edit the *Text Lookup* command on line 40 and change the algorithm:



Text Lookup comment here

Input Parameters

Language ?	en-US
Text* ?	\${vDataEntryCompany}
Data Source* ?	Data Table
Algorithm* ?	Fuzzy
Fuzzy Algorithm* ?	Levenshtein Distance
Tolerance* ?	Normal
Data Table* ?	\${vListedCustomers}
Use Column Name ?	<input type="checkbox"/>
Name* ?	CompanyAddress

Output

Text Found ?	vMatchedCompany
Index ?	vListedIndex
Success ?	success

Cancel Save

Run the script again. Is the matching better or worse? Which fuzzy matching algorithm do you think is best for this task?¹

¹ Soundex is better for matching dictated text whereas the Levenshtein algorithm is better for matching typos. This scenario is for matching dictated addresses so the Soundex algorithm would be a better choice. See https://en.wikipedia.org/wiki/Levenshtein_distance



4 Second Scenario Part 1 – Knowledge Base Creation

In this scenario we will create a knowledgebase to demonstrate AI within IBM RPA.

4.1 Scenario Description

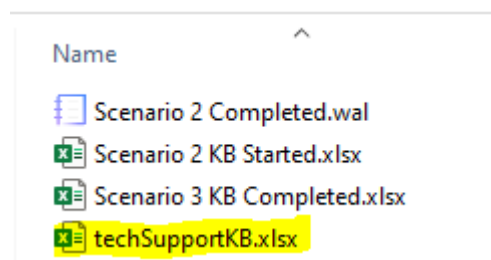
You will implement a knowledgebase to classify technical support queries.

4.2 AI - Real World Decisions

RPA bots are great at making Boolean (true/false) decisions. If a value is true bot will do X, otherwise Y. However, that scenario doesn't always reflect real life. There are many real-world decisions that are not black and white but shades of grey. A human decides to take an action based on cumulated past knowledge and not on simple true/false logic. Traditionally, 'grey' decisions required humans. But with IBM RPA's knowledgebase, a bot can classify incoming content and automatically make these decisions. In other words, bots can be cognitive.

4.3 Build Knowledge Base

Open *Lab 2 - AI with RPA\Scenario2\techSupportKB.xlsx*



Within this file there are Question, Answer, Context, and Tags:

Question will be the subject Line

Answer will be what we'd like it to take an action

Context is a topic header we'd give the cluster

Tags is just a family tag for us to track. Note the + in front of the entry.

More details on the format specified here:

<https://www.ibm.com/docs/en/rpa/21.0?topic=base-creating-knowledge-file#the-kb-spreadsheet>

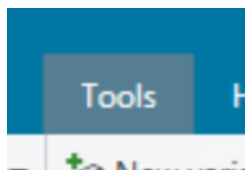


	A	B	C	D
1	Question	Answer	Context	Tags
2	I can't login	Reset Password	PasswordReset	+PASSRESET
3	I forgot my password	Reset Password	PasswordReset	+PASSRESET
4	My login is locked	Reset Password	PasswordReset	+PASSRESET
5	I'm locked out	Reset Password	PasswordReset	+PASSRESET
6	My password does not work	Reset Password	PasswordReset	+PASSRESET
7	My PC won't reboot	Reinstall PC	ResinstallPC	+REINSTALLPC
8	My laptop crashed	Reinstall PC	ResinstallPC	+REINSTALLPC
9	Fatal disk error	Reinstall PC	ResinstallPC	+REINSTALLPC
10	Blue screen	Reinstall PC	ResinstallPC	+REINSTALLPC

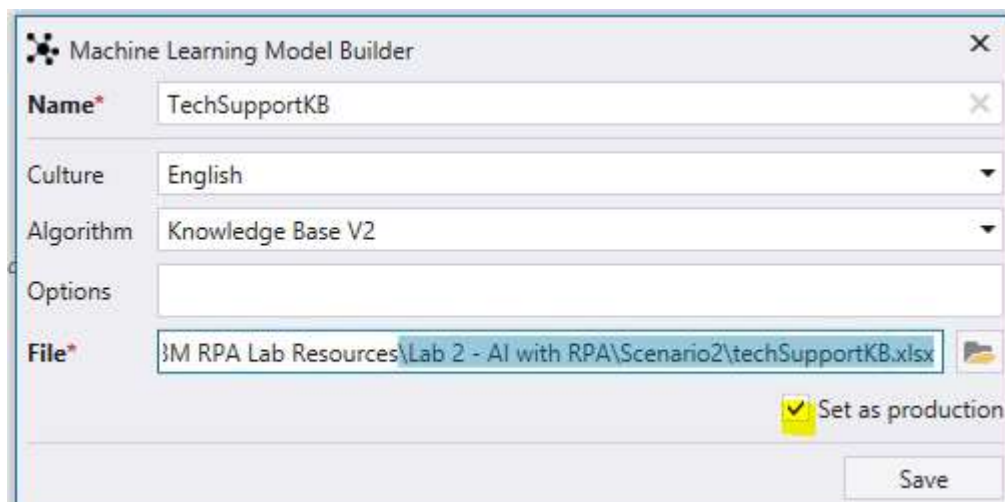
Feel free to add or edit entries, ensuring you keep the same format. When you are done, save the file and close the spreadsheet.

4.4 Upload Knowledge Base

In IBM RPA Studio, click Tools at the top of the screen and select the *Machine Learning Model Builder* option



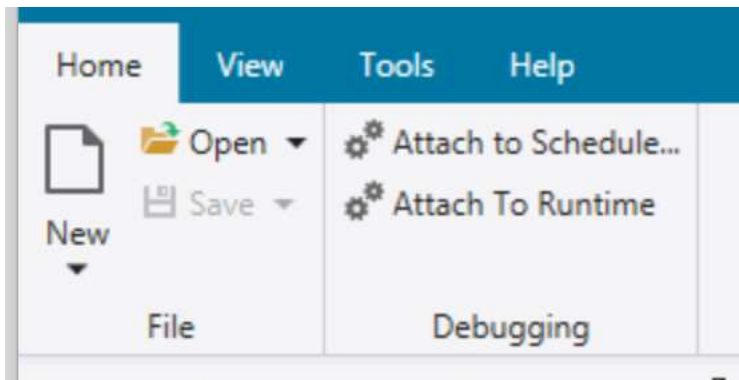
Type *TechSupportKB* as the name for the knowledge base. Leave the culture as *Default*. Select *Knowledge Base V2* as the Algorithm. Using the folder icon select the file *techSupportKB.xlsx*. Click *Set as production* and save. See below.





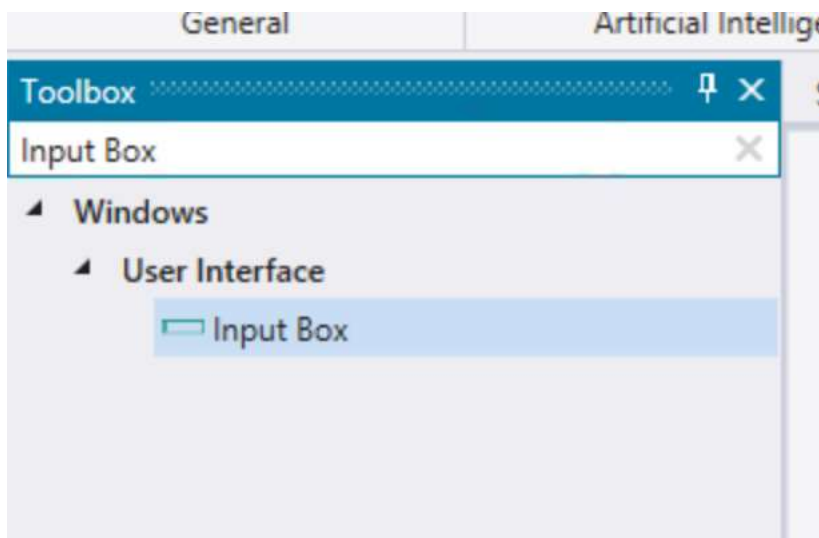
4.5 Build Script

Select *Home*, then *New* in the top left. Select *WAL File*



4.5.1 Input Box Command

Search the toolbar for an “Input Box” command



Drag the command to your palette

Provide the following fields:

Title: Ask a Question

Prompt: What is your tech support question?

Text: vUserInput

Note: you need to create *vUserInput* as a new text variable



Input Box

comment here

Input Parameters

Title* ? Ask a Question

Prompt* ? What is your tech support question?

Default Value ?

Output

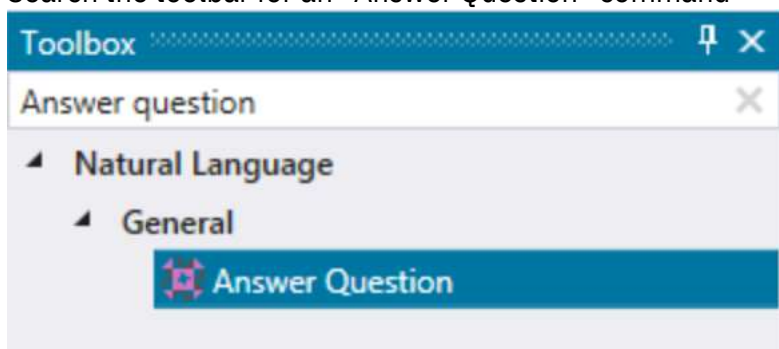
Text ? \${vUserInput}

Cancel Save

Click Save

4.5.2 Answer Question Command

Search the toolbar for an “Answer Question” command



Drag the command to your palette under the first command

Provide the following parameters to the input fields. Note when entering *Knowledge Base* you should select your uploaded KB from the previous step.



Answer Question

comment here

Input Parameters

Language ? en-US

Text* ? \${vUserInput}

Knowledge Base* ? TechSupportKB

Version ?

Minimum Score ?

Answers Quantity ?

Options ?

Bot Ask History ?

Output

Answer ? \${vResponse}

Tags ?

Context ?

Score ?

Additional Answers ?

Data ?

Success ?

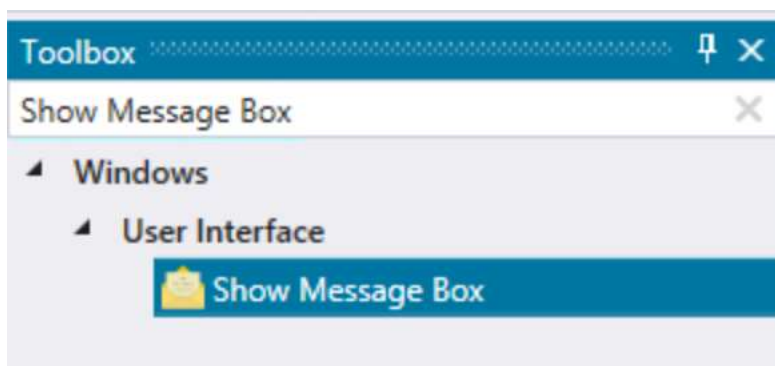
Cancel Save

Note: you need to create *vResponse* as a new text variable.

When done, click Save.

4.5.3 Show Message Box Command

Search the toolbar for a “Show Message Box” command



Drag the command to your palette under the second command

Provide the following inputs:



Title: RPA AI Response

Text: Your bot believes that \${vUserInput} should followup the action with \${vResponse}

comment here

Input Parameters

Title* ? RPA AI Response

Text* ? Your bot believes that \${vUserInput} should followup the action with \${vResponse}

Icon* ? Information

Buttons* ? OK

Default button* ? First button

Output

Result ?

Cancel Save

Click Save

4.6 Run Script

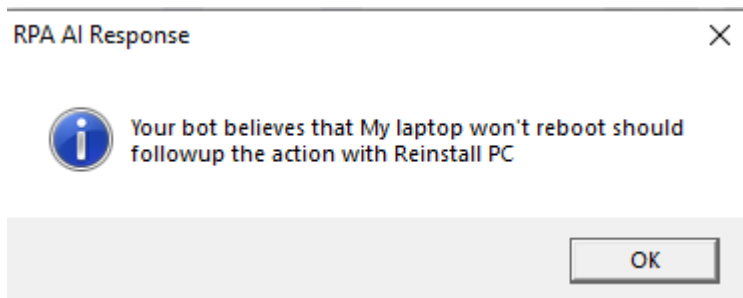
Run the script with Ctl+F5. Enter the following question:

Ask a Question

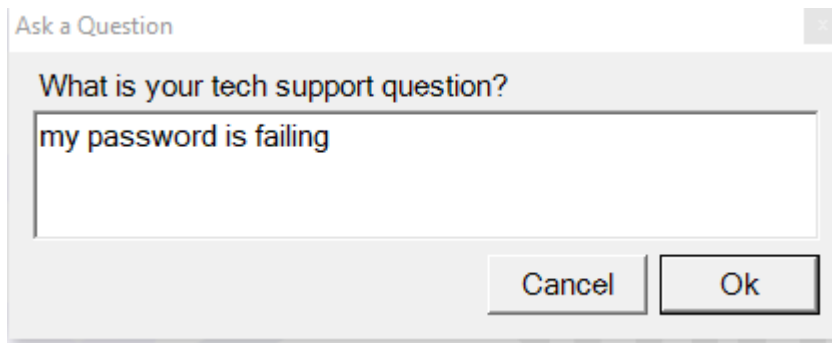
What is your tech support question?

My laptop won't reboot

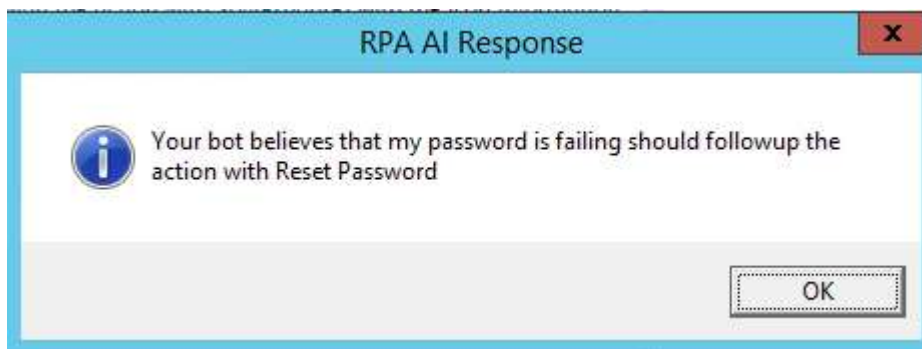
Cancel Ok



Now run the script again. This time enter the following:



This time the response is:



Note that both questions entered were not an exact match of any of the questions entered in the knowledge base. RPA uses machine learning to classify the question to give the most appropriate answer. This is AI in action!



5 Second Scenario Part 2 – Knowledge Base Training

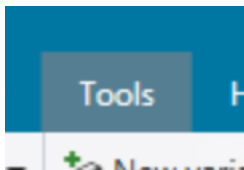
In this scenario, we will train the knowledge base.

5.1 Real World Alignment

Updating a knowledge base spreadsheet can be time consuming and error prone. To make things easier, you can train the knowledge base within RPA studio.

5.2 Navigate to Knowledge Base Training

Click Tools at the top of the screen and select the Knowledge Base Training option



5.3 Open the Knowledge Base

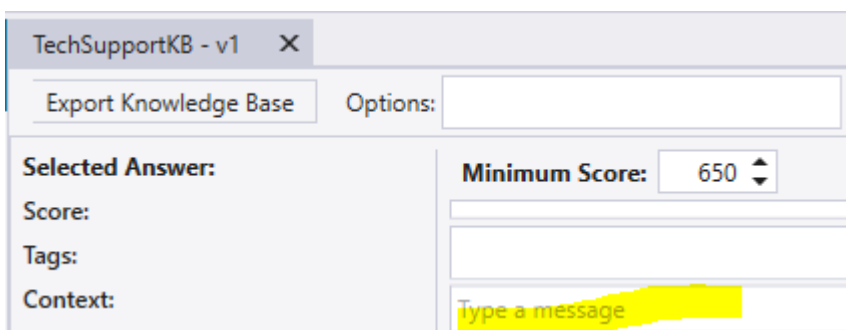
You will see *Knowledge Bases* pop up on the left side. Within this tab, find *TechSupportKB* and select Version 1. See below:



5.4 Training the Knowledge Base

5.4.1 Add new classification

In this exercise you will add a new classification. Within RPA Studio, find the text box containing the faint grey text “*type a message*”:





Replace “Type a message” with “My laptop is on fire” and hit enter.:

TechSupportKB - v1 X

Export Knowledge Base Options:

Selected Answer:

Score:

Tags:

Context:

Minimum Score: 650

You should see the following (incorrect) response from the knowledge base:

Reinstall PC edit | select

Reinstall PC edit | select

Reset Password edit | select

Reset Password edit | select

Add New Answer:

Add New Answer +

User: My laptop is on fire

Bot: Reinstall PC ⚠

Type a message

We need to retrain the knowledge base with a new answer. On the left side of the Knowledge Base, add a new answer “Use a fire extinguisher”:

Reinstall PC edit | select

Reinstall PC edit | select

Reset Password edit | select

Reset Password edit | select

Add New Answer:

Use a fire extinguisher +

User: My laptop is on fire

Bot: Reinstall PC ⚠

Type a message

Select the + button to add the answer to the knowledge base. The *Add New Answer* panel should appear. Select + *Add new* and enter a new context of *PutOutFire*. You should see the following:



Add New Answer

×

Answer Text:

Use a fire extinguisher

Tags:

+

Context:

PasswordReset

ResinstallIPC

PutOutFire

+ Add New

PutOutFire

Cancel

Save

Now select the + to the right of the Tags Text box:

Add New Answer

×

Answer Text:

Use a fire extinguisher

Tags:

+

Context:

PasswordReset

ResinstallIPC

PutOutFire



The *Tags* panel should appear. Enter a tag of *+PutOutFire*. Note the + symbol must be the first character.

Tags

Tag Text:

+PutOutFire

Cancel Save

You should see the following:

Add New Answer

Answer Text:

Use a fire extinguisher

Tags:

+PutOutFire

Context:

- PasswordReset
- ResinstallPC
- PutOutFire
- New Context

+ Add New

PutOutFire

Cancel Save

Press *Save*. And then *Save* again.

Click “*Retrain*” in the top right corner (see below). This will update the knowledgebase.



Export Knowledge Base

Options:

Retrain

Minimum Score: 650

Related Questions:

My laptop is on fire

User:

My laptop is on fire

Bot:

Use a fire extinguisher

5.4.2 Test new classification

Inside the “Type a message” field, enter “*My laptop is on fire*”. This time the answer is successfully classified as “Use a fire extinguisher”.

User:

My laptop is on fire

Bot:

Use a fire extinguisher



5.4.3 Optional extras

Add a couple of additional fire related tech support questions such as "There's smoke coming out of my laptop" and "I can see some flames under my keyboard"

5.4.4 Reclassification

In this exercise you will continue working with the same knowledgebase, but this time reclassify an existing sentence.

Find the text box containing *"type a message"*:

The screenshot shows a web interface for 'TechSupportKB - v1'. It has a tab labeled 'Export Knowledge Base' and an 'Options:' field. Below this, there are fields for 'Selected Answer:', 'Score:', 'Tags:', and 'Context:'. To the right of these fields is a 'Minimum Score:' field set to '650'. In the 'Context:' field, the text 'Type a message' is highlighted in yellow.

Replace *"Type a message"* with *"I cannot login to my laptop"* and hit enter.:

The screenshot shows a chat interface with a blue background. The 'User:' message is 'I cannot login to my laptop'. The 'Bot:' response is 'Use a fire extinguisher' with a yellow warning triangle icon next to it.

The AI engine thinks the question *"I cannot login to my laptop"* most closely maps to the Answer *"Use a fire extinguisher"*. This is the wrong answer - It should be *"Password Reset"*. We must retrain it.

On the left side, click *Select* on one of the *"Reset Password"* entries, then click *"Retrain"* in the top right corner (see below). This will update the knowledgebase.



Knowledge Base Options: Retrain

Related Questions:

My PC won't reboot

My laptop crashed

Blue screen

Fatal disk error

I cannot login to my laptop

5.4.5 Test reclassification

Inside the “Type a message” field, enter “*I cannot login to my laptop*”. This time the answer is successfully classified as “*Reset Password*”.

User:

I cannot login to my laptop

Bot:

Reset Password



5.4.6 Save changes

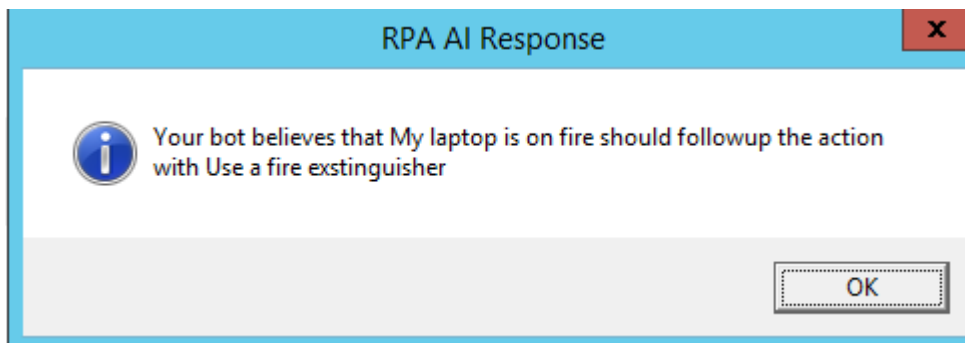
To persist your changes, click on the *Changes* tab at the bottom of the knowledge base panel, click *Production Version* check box and press the *Publish*:

Date	Question	Answer	Type	
8/7/2021	I cannot login to my laptop	Reset Password	Conhecimento Adicionado	X

☒ Production Version **Publish**

Trainer Terms Synonyms **Changes**

The knowledge base is now saved on your tenant where any bot can use it. To prove, close the Knowledgebase Editor and re-rerun the bot as described in section 4.6. This time if you type “My laptop is on fire” you will get the response:





6 Third Scenario - Text Classifier

In this scenario, we will train a text classifier. A text classifier differs from a knowledge base. It classifies text into categories which RPA can use to make decisions.

6.1 Scenario Description

You will build a bot for a restaurant to automatically respond to good and bad reviews based on the classification of the review.

6.2 Inspect File

Open the folder *Lab 2 - AI with RPA\Scenario 3*. Unpack the *CustomerReviews.zip*. Within the unzipped folder you will see a folder called *TextClassifier* and under this, two folders containing good and bad reviews. Examine some of the reviews.

6.3 Open a work in progress script

Open the folder **Lab 2 - AI with RPA\Scenario3**. Open IBM RPA Studio and select *Open* in the top left. Open *RestaurantReviewStarted.wal*. Your script will open in Studio.

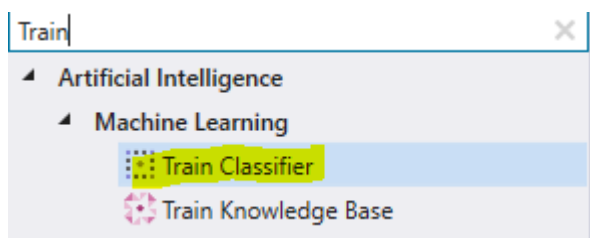
Examine the script. You will see a loop to extract reviews from customers and then respond. The code does not yet contain AI. Your task is to train and then add the text classifier.

6.4 Train the Text Classifier

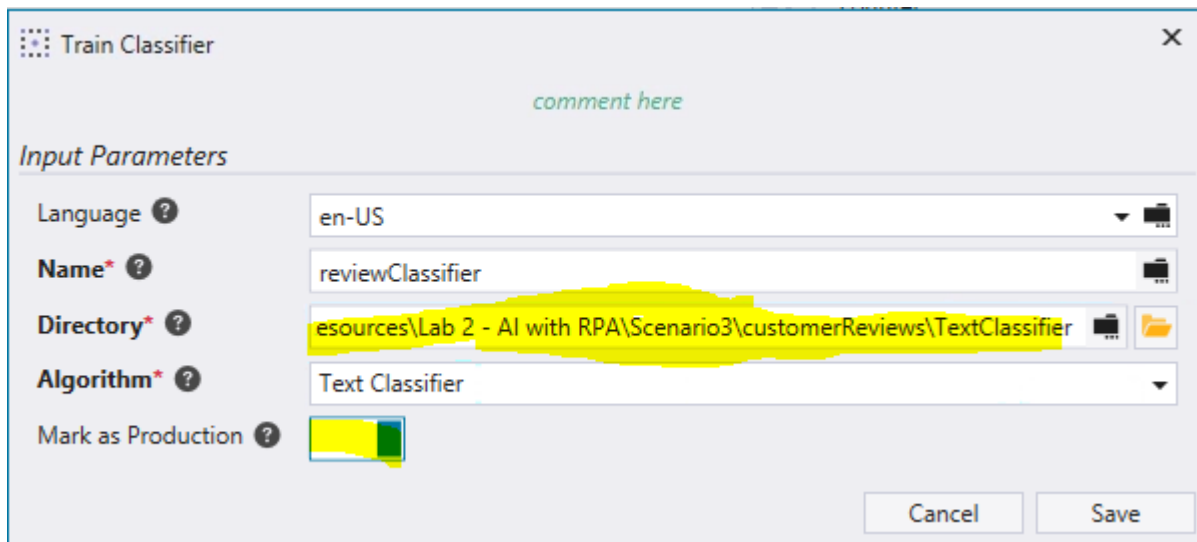
On line 7 you will see a TODO:

7 *TODO: Train the classifier here*

Search the toolbar for the “Train Classifier” command:



Drag the command to line 7 of your script. Edit this command to provide the following fields:



The 'Train Classifier' dialog box is shown. It has a title bar with a close button. Below the title bar is a green text prompt 'comment here'. The 'Input Parameters' section contains the following fields:

- Language: en-US
- Name: reviewClassifier
- Directory: esources\Lab 2 - AI with RPA\Scenario3\customerReviews\TextClassifier (highlighted in yellow)
- Algorithm: Text Classifier
- Mark as Production: ☒ (highlighted in yellow)

At the bottom right are 'Cancel' and 'Save' buttons.

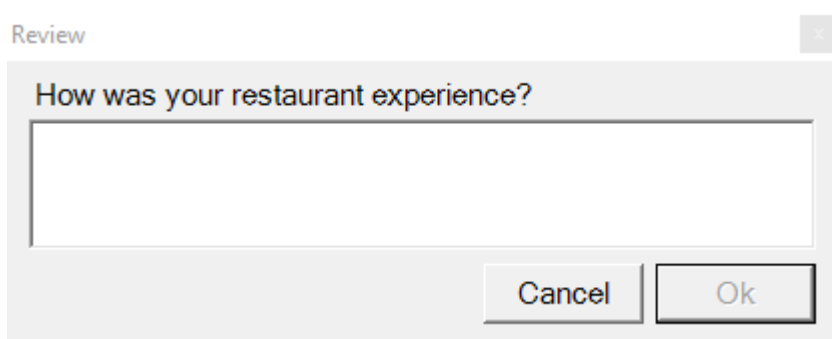
NOTE: When setting *Directory*, specify folder:

Lab 2 - AI with RPA\Scenario3\customerReviews\TextClassifier

Ensure *Mark as Production* is enabled. Press *Save*.

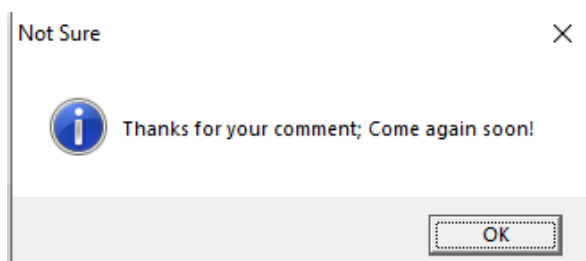
Now run the bot. You will see the bot pause for several seconds whilst it reads and classifies the reviews and then stores it in the tenant.

After a while you should see the following dialog appear four times:



The 'Review' dialog box is shown. It has a title bar with a close button. The main text is 'How was your restaurant experience?'. Below this is a large text input field. At the bottom are 'Cancel' and 'Ok' buttons.

Enter two good reviews and two bad. You will see the same response each time:



The 'Not Sure' dialog box is shown. It has a title bar with a close button. The main text is 'Thanks for your comment; Come again soon!'. Below this is an 'OK' button.

It's the same response because we have not yet added the text classifier command. We will do this in the next step!



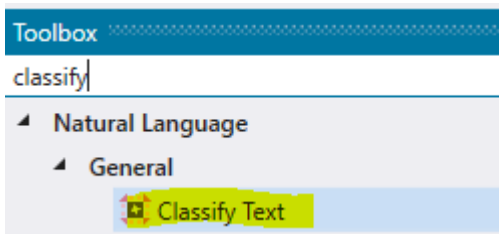
Click Cancel in the Text Input, and then press STOP to exit.

6.5 Add the Text Classifier

In this step, you will add the text classifier command. Above the *Log Message* command, you will see a TODO:



Search the toolbar for the “Classify Text” command:



Drag the command under the TODO of your script. Provide the following fields:

Classify Text

comment here

Input Parameters

Language ? en-US

Text* ? \${response}

Model* ? reviewClassifier

Version ?

Output

Best Choice ? \${textBestChoice}

Best Choice Score ? \${bestChoiceScore}

Label and Score ? \${labelAndScore}

Success ?

Cancel Save

NOTE: Use the variables already defined, and when setting the Model, make sure it is the same as the one entered in 6.4.

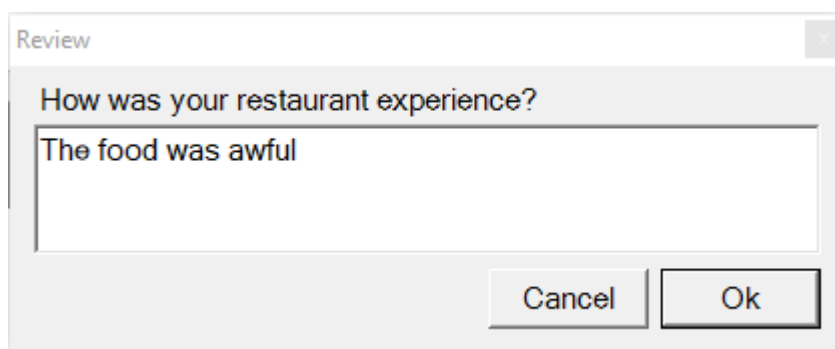


6.6 Run the Script

Comment out *trainClassifier*. We do not need to run this command again unless the training files are regularly updated. You can also train the model using the Machine Learning Model Builder as demonstrated in section 4.3.

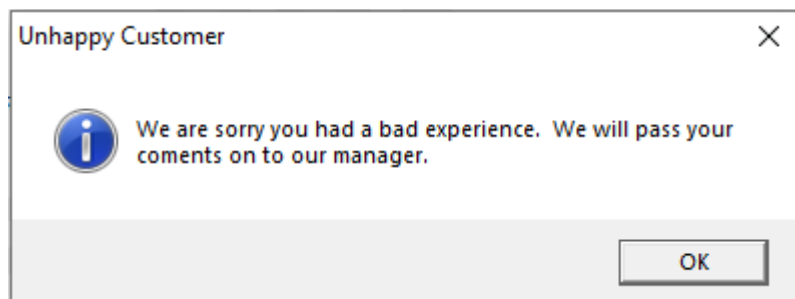
```
trainClassifier --folder "D:\RPA\AI_ENABLEMENT\LABS\RPAAdvancedSept2021\Lab 1 - AI with RPA\Scenario3\customerReviews\TextClassifier" --algorithm "TextClassifier" --culture "en-US" --name reviewClassifier --production
```

Run the script by pressing Ctrl+F5 (run without debugging).
You should see the same dialog box appear as before. Enter a bad review:



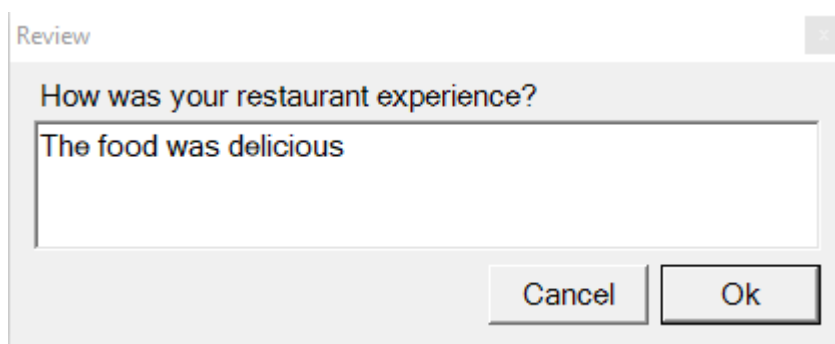
A dialog box titled "Review" with a close button in the top right corner. It contains the question "How was your restaurant experience?" followed by a text input field containing the text "The food was awful". At the bottom right, there are two buttons: "Cancel" and "Ok".

The review is correctly identified as bad:



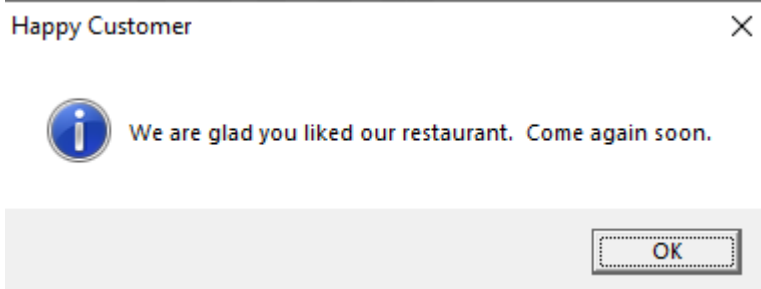
A dialog box titled "Unhappy Customer" with a close button in the top right corner. It features an information icon (a blue circle with a white 'i') on the left. To the right of the icon, the text reads: "We are sorry you had a bad experience. We will pass your comments on to our manager." At the bottom right, there is an "OK" button.

Now enter a good review:

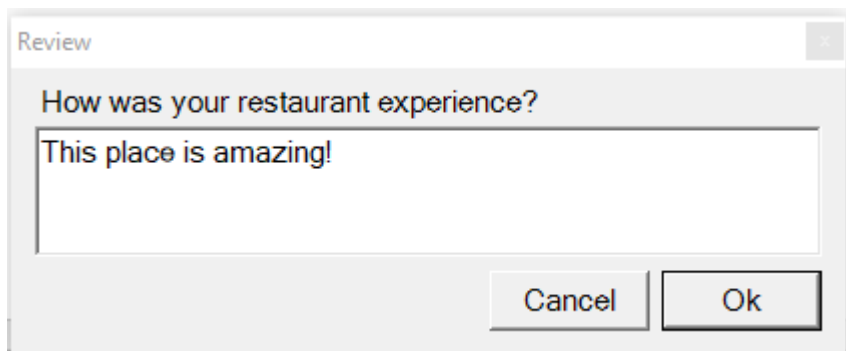


A dialog box titled "Review" with a close button in the top right corner. It contains the question "How was your restaurant experience?" followed by a text input field containing the text "The food was delicious". At the bottom right, there are two buttons: "Cancel" and "Ok".

The bot classifies the review and responds accordingly.



Enter a review that is incorrectly classified.



It incorrectly classifies it as a bad review. Edit the test responses to add this comment and re-run the classifier.



7 Fourth Scenario – Basic R Script

R is an open-source language for [statistical computing](#). In this scenario, we will integrate R with WAL script for statistical analysis.

A prerequisite for running R is to install it. If it not already installed, the latest compatible version is here:

Install R V3.4.4 found here:

<https://cran.r-project.org/bin/windows/base/old/3.4.4/>

Use all default settings when installing.

Note: newer versions of R are not supported natively within IBM RPA. Once you have installed R, you need to restart RPA Studio for it to pick up the R install location.

7.1 Scenario Description

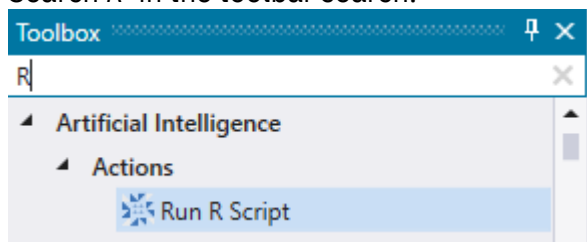
The Focus Corp sales director needs to track the performance of her tele-sales team. She requires a button on her laptop to chart her top sellers. You are an automation developer responsible for implementing this.

7.2 Open a work in progress script

Open folder **Lab 2 - AI with RPA\Scenario4**. Open IBM RPA Studio and select *Open* in the top left. Go into *Scenario 4* folder and open *Scenario 4 Started.wal*. Your script will open in Studio.

7.3 Run the R Script

Search *R* in the toolbar search.

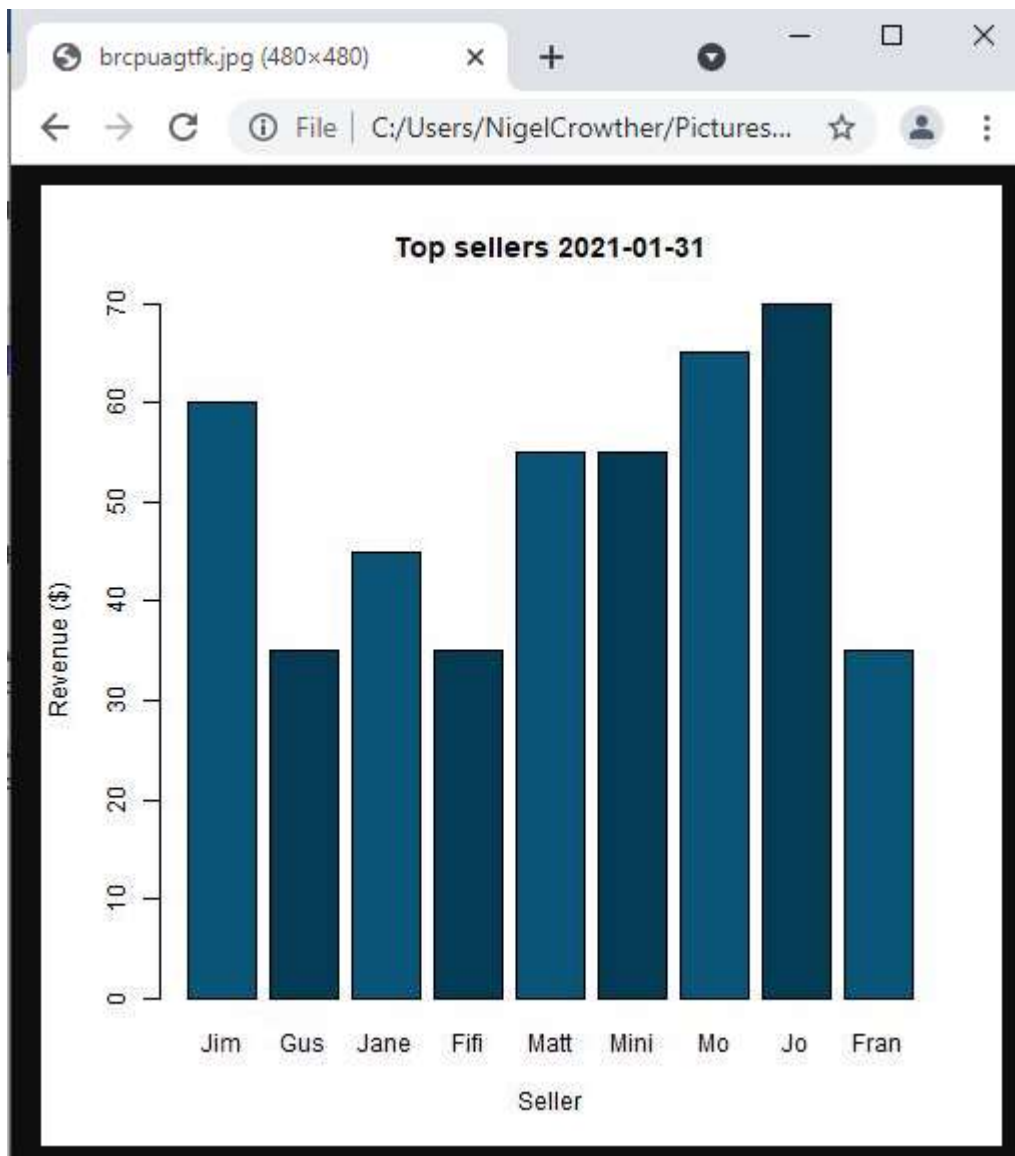


Drag the command to line 20 and paste in the following R code:

```
sellers <- c(${sellers})
quantities <- c(${quantities})
myPicturesPathR <- ${myPicturesPathR}
pictureName <- ${pictureName}
barplot(quantities, main=paste("Top sellers ",${currentDate},sep=""), xlab="Seller",
ylab="Revenue ($)", names.arg=sellers, col=c("#0a5477","#053b54"), ylim=c(0,70))
dev.copy(jpeg,filename=paste(myPicturesPathR,"\\",pictureName,".jpg",sep=""))
dev.off()
```



Press Save. Now Run the script. You should see the following graph appear in your browser:



7.4 If you have time

Set up the bot so that it can be launched as an attended bot



8 Fifth Scenario – Machine learning with R

In this advanced lab you will apply machine learning to multivariate travel data.

8.1 Scenario Description

A travel company needs to improve the hit rate offered to customers. In other words, offer customers holidays based on the choices made by customers in the past. In this lab you will implement a predictive model using R and then call this model from a chat bot.

8.2 Install latest version of R

Install the latest version of R found here:

<https://cran.r-project.org/bin/windows/base/>

Use all default settings when installing. When installed, restart IBM RPA Studio so that it picks up the new R install.

8.3 Create the predictive model

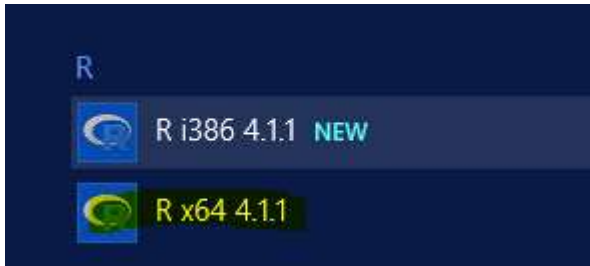
Open **Lab 2 - AI with RPA\Scenario5\BrowsingCruises.csv**

You should see the following data (truncated)

	A	B	C	D	E	F
1	age	job	marital	education	Income	y
2	56	housemai	married	basic.4y	12500	no
3	57	services	married	high.scho	34000	yes
4	37	services	married	high.scho	40000	no
5	40	admin.	married	basic.6y	18000	no
6	56	services	married	high.scho	20000	no
7	45	services	married	basic.9y	12900	no
8	59	admin.	married	profession	32300	no
9	41	blue-colla	married	unknown	48000	no
10	24	techniciar	single	profession	18000	no
11	25	services	single	high.scho	26000	no
12	41	blue-colla	married	unknown	12500	no
13	25	services	single	high.scho	34000	no
14	29	blue-colla	single	high.scho	40000	no
15	57	housemai	divorced	basic.4y	18000	yes

This data represents customers that either booked a cruise (y = yes), or declined a cruise (y = no)

Now let's use this data to create a predictive model.
Start an R Console 4.1.1 x64 (or later)



Paste the following script within the R console:

```
install.packages("caret")
install.packages("e1071")

library(caret)

setwd(choose.dir(caption = "Select the folder with the data"))

data = read.csv("BrewsingCruises.csv")

model = train(y ~ .,
              data,
              method = "glm",
              family = "binomial")

varimp <- varImp(model)
plot(varimp, main="Variable Importance")

saveRDS(model, "model.rds")

print("Model Saved..")
```

This script performs the following:

1. Prompts for a mirror location. Enter the location nearest to you. This installs the machine learning package called *caret* from that location
2. Requests you to supply the folder in which the ML spreadsheet resides. Select the **c:\Users\Administrator\Desktop\IBM RPA Lab Resources\Lab resources for AI and OCR\Lab 2 - AI with RPA\Scenario5** folder
3. Creates a predictive model from the data in *scenario5/BrewsingCruises.csv*
4. Prints a chart showing relative importance of each variable.
5. Saves the model as *model.rds* in the same folder. The model contains customer predictions from *BrewsingCruises.csv*

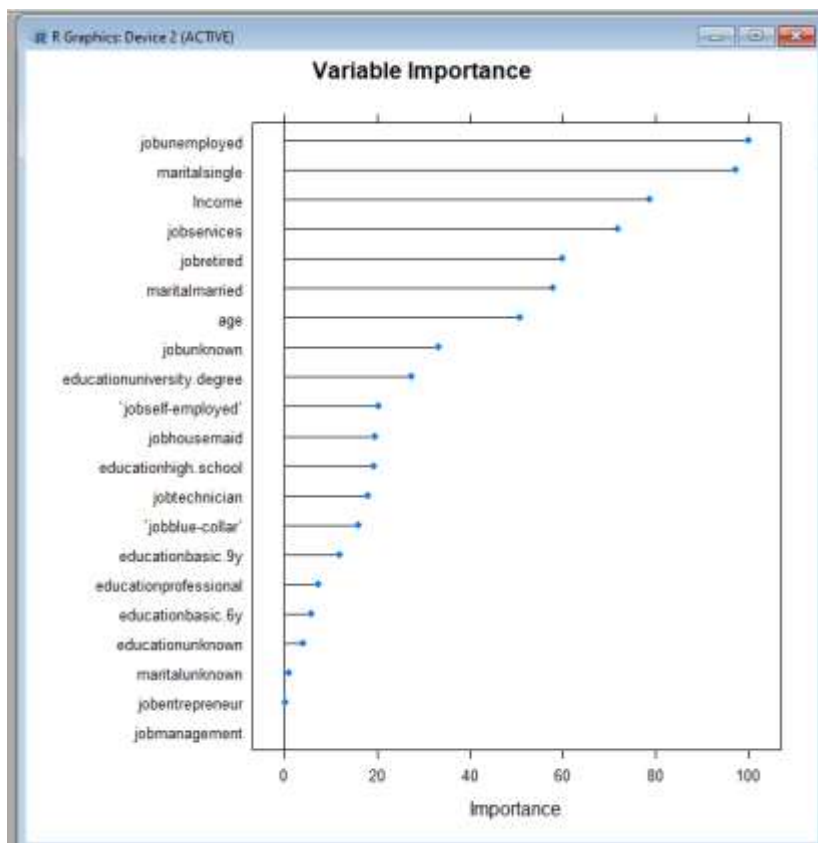
When the script has run successfully you should see the following in your console. Note there should be no errors, and several warnings.:



```
R Console

> model = train(y ~ .,
+               data,
+               method = "glm",
+               family = "binomial")
There were 50 or more warnings (use warnings() to see the first 50)
>
> varimp <- varImp(model)
> plot(varimp, main="Variable Importance")
>
> saveRDS(model, "model.rds")
>
> print("Model Saved..")
[1] "Model Saved.."
> |
```

You should see a chart showing the importance of each variable:





8.4 Open a work in progress script

Open IBM RPA Studio and select *Open* in the top left. Go into **Lab 2 - AI with RPA\Scenario5** and open *Scenario 5 Started.wal*. Your script will open in Studio. Open line 8 containing the *Run Windows Script* command. In the Code section, paste the following, replacing the PATH and SCRIPT_HOME variables with the correct paths for your system:

```
@echo off

REM run r with params %1

IF %1.==. GOTO No1

set PATH=%PATH%;c:\Program Files\R\R-4.1.1\bin
set SCRIPT_HOME="C:\WDG\RPA Advanced\Lab 2 - AI with RPA\Scenario5"

rscript %SCRIPT_HOME%\PotentialCustomer.r %SCRIPT_HOME% %1

GOTO End1

:No1
    ECHO No param 1
GOTO End1

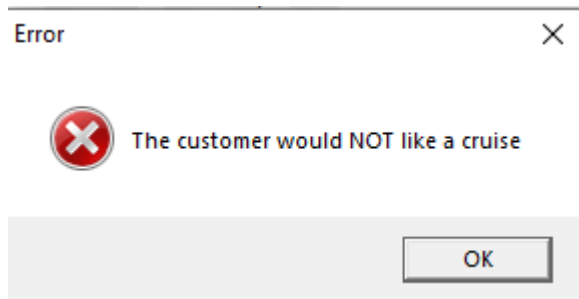
:End1

echo %result%
```



8.5 Run the R Script

Press Save and Run the script. After a several seconds you should see the following appear in your browser:



What happened? The script read *Senario5/CruiseCustomerNo.csv* containing the following data:

1	age	job	marital	education	Income
2	56	housemaid	married	basic.4y	12000

It passed this data to an R script invoked from a DOS command. The R script references the model we created earlier to base a prediction. The script returns either *Yes* or *No* and pops up a message with the appropriate response. The script is called *PotentialCustomer.r* and it is shown below:

```
# test if the args were passed; if not, return an error
args = commandArgs(trailingOnly=TRUE)
if (length(args) < 2) {
  stop("Two arguments must be supplied (path and customerdata).n", call.=FALSE)
}

## arg 1 contains path of the csv file and the model
setwd(args[1])

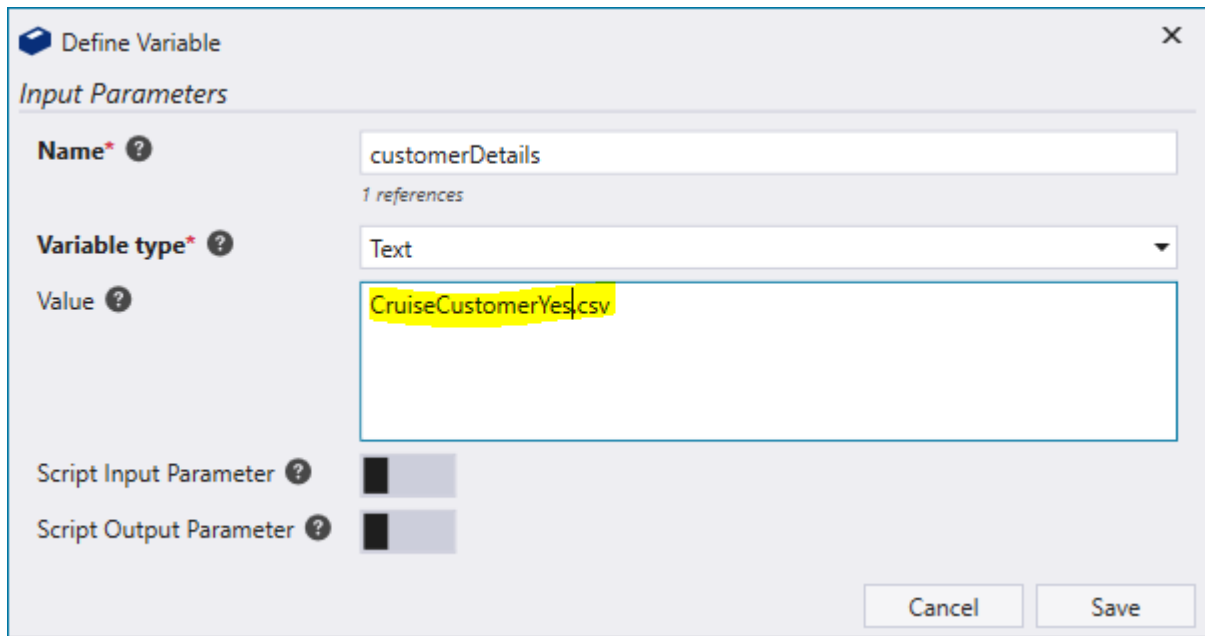
## Read the predictive model
model <- readRDS("model.rds")

## Read the csv file supplied as the second argument
customer = read.csv(args[2])

## Invoke prediction using Caret library
p = predict(model, customer)

## return result (Y/N) as string which RPA will parse
print(p)
```

Now edit *line 6 of Senario5Started.was* and change the input file to *CruiseCustomerYes.csv*. See below:

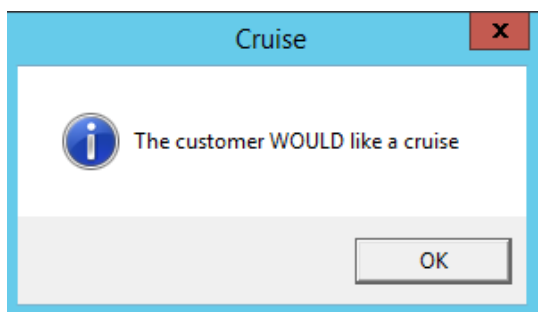


The 'Define Variable' dialog box is shown with the 'Input Parameters' tab selected. The 'Name' field contains 'customerDetails', the 'Variable type' is set to 'Text', and the 'Value' field contains 'CruiseCustomerYes.csv'. Below these fields are two checkboxes: 'Script Input Parameter' and 'Script Output Parameter', both of which are unchecked. 'Cancel' and 'Save' buttons are at the bottom right.

Press Save. Examine the contents of the csv file. You should see it contains a potential customer who is likely to book a cruise given the machine learning model:

age	job	marital	education	Income
57	services	married	professional	74000

Run this script again. This time you should see the message:



What are the advantages and disadvantages of using R Machine Learning over using the built in Knowledge bases within RPA? The advantages are that you can build complex multivariate models and train the model with popular data science packages such as Caret. Disadvantages are that the model is stored on a local drive, not in the tenant

NOTES

1: The reason we invoke R from DOS rather than with the native R command is that the latest version is required and that is not supported in IBM RPA.

2. We are just scratching the surface of R capabilities. Take a look at the Caret library for more details.



9 Sixth scenario – Business Rules

In this scenario you will build a decision table using RPA commands.

9.1 Scenario Description

The Focus Corp wants to offer discounts to customers who purchase in bulk. The following decision table is used:

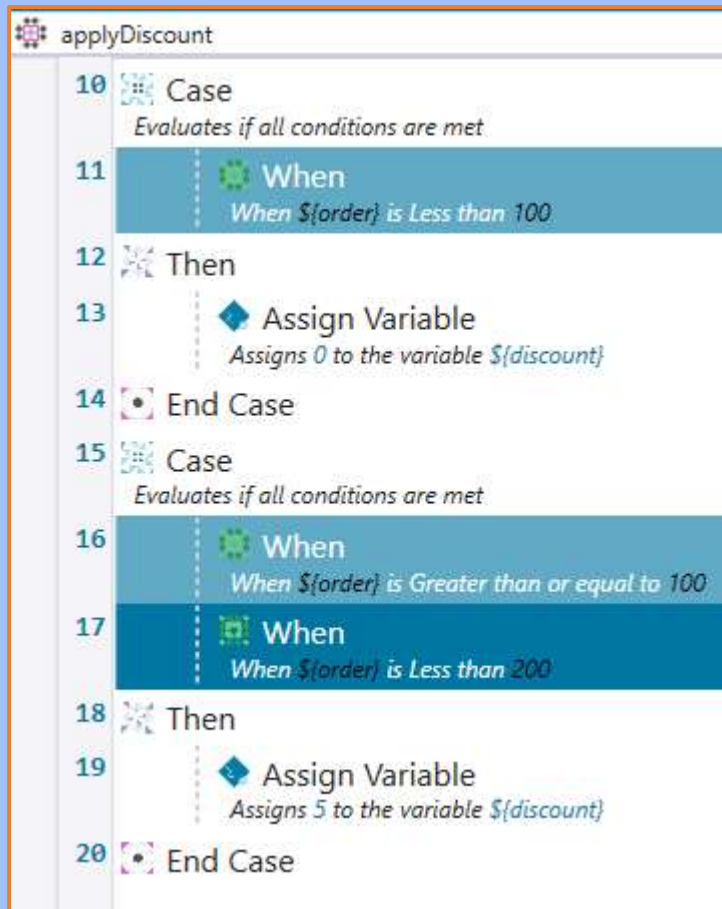
Inputs		Outputs	
Number of units		Discount	
Number		Percentage	
<	100		0.00 %
<	200		5.00 %
<	500		10.00 %
<	1000		15.00 %
≥	1000		20.00 %

You are tasked with implementing this table in WAL.

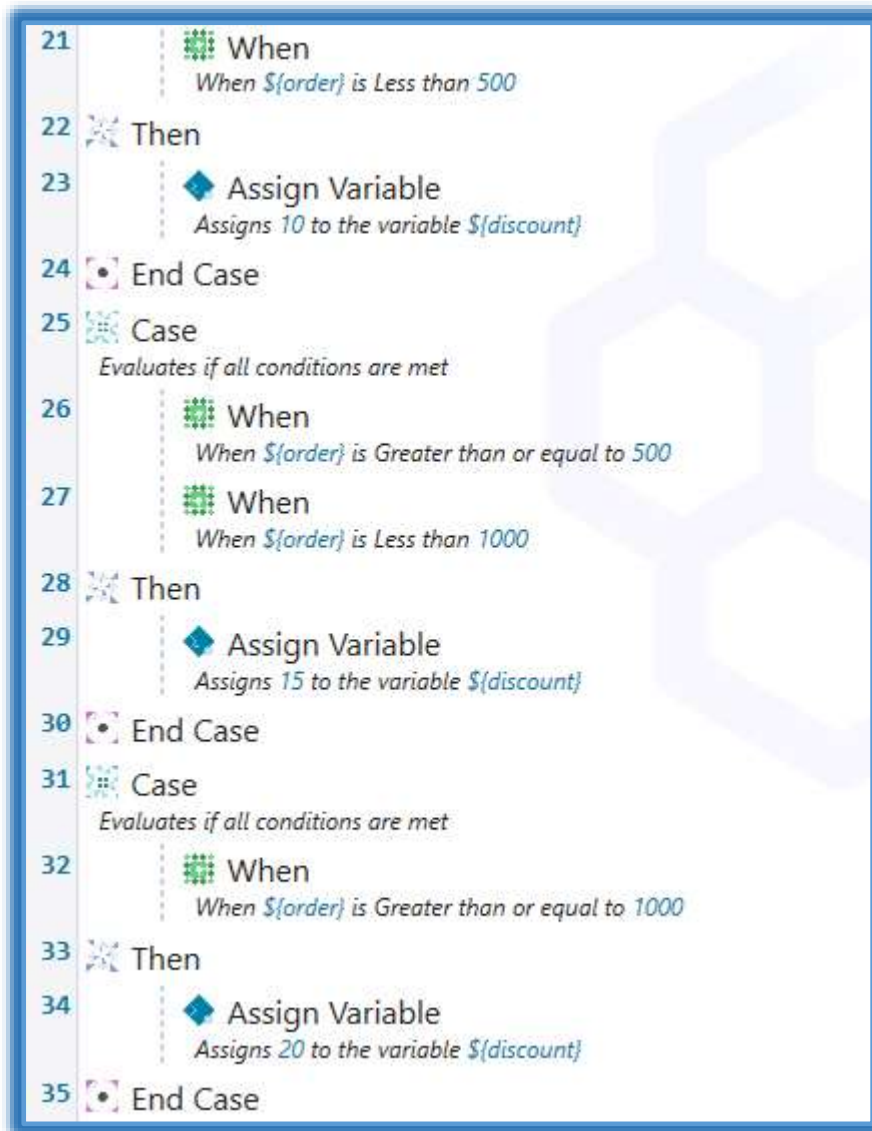
9.2 Open a work in progress script

Open the folder **Lab 2 - AI with RPA\Scenario6**. Open IBM RPA Studio and select *Open* in the top left. Open *BusinessRulesInProgress.wal*. Your script will open in Studio.

Examine the script. Drill into to the *applyDiscount* rules. You should see the following rules which implement the first two rows of the decision table above.



Add the additional three rules:



9.3 Test bot

Now go to line 1 and set the order amount variable to a value that will hit one of the rules:

1 Define Variable

Define a variable with name order, type Numeric and default value 1000

Run the bot. in the above example, the order number was 1000 so it hits the rule on line 32 resulting in a discount of 20%



Congratulations! You have completed the lab.