



kitewheel

orchestrate great experiences

Kitewheel Decisioning

The slide features a dark blue background with a diagonal light blue stripe. Several dandelion seeds are scattered across the frame, some in sharp focus and others blurred. The text 'Training Overview' is centered in white.

Training Overview

Kitewheel Decisioning Learning Outcomes

- ◀ What decisioning options are available?
- ◀ How do I decide which node to use?
- ◀ Using Basic Nodes
 - ◀ Conditional
 - ◀ 2D Table
 - ◀ Decision Tree
 - ◀ Columnar Table
- ◀ Using Script Nodes
 - ◀ JavaScript
 - ◀ R

Target Audiences

Primary

Journey Manager



Joanna, the Journey Manager

INTERESTED IN...

- Validate and challenge pre-defined notions of customer journey
- Design, measure and investigate use cases (or visions)
- ROI
- Impact on customer retention, acquisition, revenue

Configurer



Cory, the Configurer

INTERESTED IN...

- Identifying data sources
- Designing logic
- Solution architecting logic and rules
- Testing and deploying new orchestration journeys

Analyst



Alex, the Analyst

INTERESTED IN...

- Discover and investigate the Who, How, Why in order to:
 - Measure and increase effectiveness
 - Predict behavior
- Map journey performance against KPIs and benchmark

Decisioning



Decisioning Nodes

- ◀ Which one is the best depends on the complexity of the decision that is being handled
- ◀ Complexity = number of inputs and outputs and whether simple conditions are sufficient



2D Table



Decision Tree



Columnar Table



JavaScript



Conditional



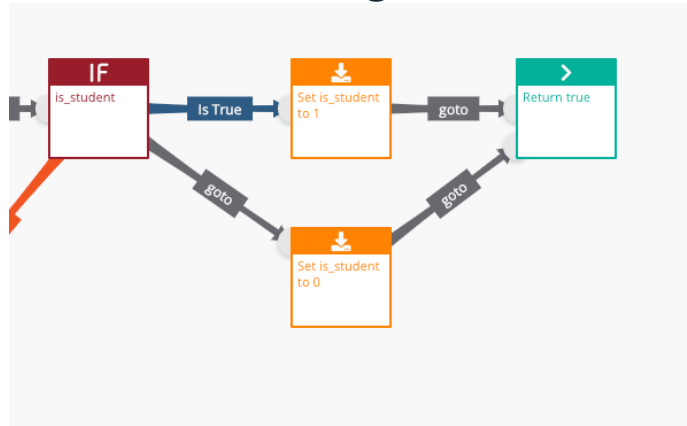
R Script

Boolean Logic – The basis of all rules

- ◀ In its simplest format ALL decision logic, regardless of the construct being used, breaks down to **TRUE** or **FALSE**
- ◀ This is called Conditional Boolean Logic
 - ◀ IF **CONDITION IS TRUE** THEN 1 [ELSE 2]
- ◀ Often this most simple conditional rule format is sufficient for what is trying to be achieved
- ◀ **CONDITIONS** can usually be Basic or Advanced expressions
- ◀ Advanced expressions are JavaScript expressions
- ◀ The “magic” variable **VAL** is always available

Conditional Logic – Example

- ⏪ If an email contains .edu then I want to set is_student to 1
- ⏪ If an email does not contain .edu then I want to set is_student to 0
- ⏪ Decision Node inputs are selected in the graph as this encourages re-use of the node



Conditional Editor ?

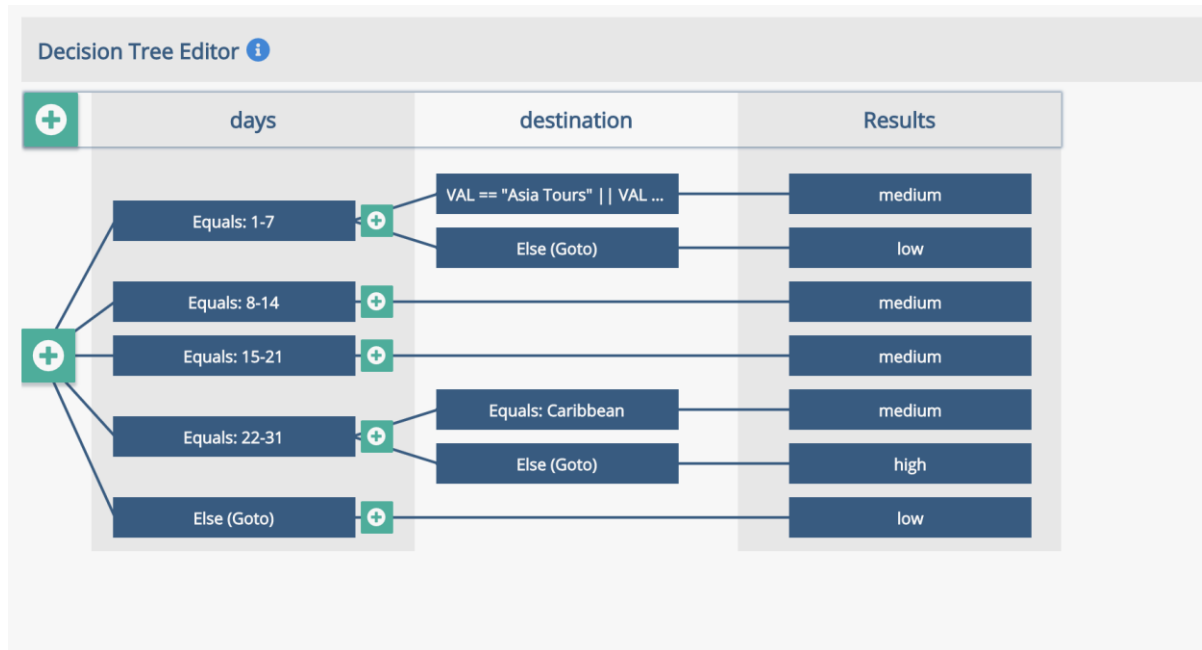
+ Create Condition

IF of the following are true:

<input type="text" value="email"/>	<input type="text" value="Contains"/>	<input type="text" value=".edu"/>	<input type="button" value="No Unsaved Changes"/>
------------------------------------	---------------------------------------	-----------------------------------	---

THEN ...return *TRUE*, otherwise return *FALSE*

Conditional Logic – Decision Tree

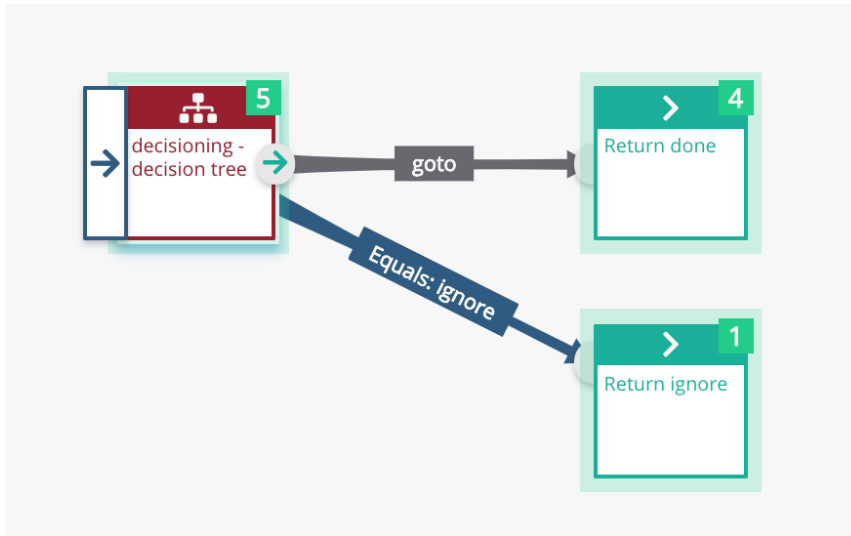


- ⏪ Segment Users based on the number of days they plan to travel and the destination they are planning on going
- ⏪ Set low priority to anything that doesn't meet criteria

Decisions Nodes Provide Outputs



- Decision Trees, Columnar Tables and 2D Tables will provide outputs to the graph



Edit 'lead_value'



↔ Replace Node

📁 Open in Decision Tree Editor

Decision Tree Return Value (optional):



days:



destination:



2D Tables are for Two Primary Attributes



X-Value: numFollowers

Y-Value: numFollowing

- Create an offer matrix based on the following / follower bands
- Take into account that people follow more than they are followed
- Choose offers for each intersection point

2D Tables are for two Primary Attributes



		X-Value i		
		followers < 10	followers < 100	Else (Goto)
Y-Value	+ Create Column			
	+ Create Row			
	following < 100	Offer A	Offer C	Offer B
	following < 1000	Offer C	Offer C	Offer C
Else (Goto)	No Offer	Offer B	No Offer	



- ◀ Used when the number of inputs ≥ 2
- ◀ Not all combinations of inputs are interesting – decisions are sparse – compared to the decision tree
- ◀ Provides one or more outputs
- ◀ Executed from left to right
- ◀ First rule that is “true” provides the output
- ◀ Blank cells are always true
- ◀ Ensure there is a catch-all rule

Columnar Table



Columnar Table Editor ?

+ Add Decision	single	couple	family	large_group
Rules +				
num_travelers	Number(VAL) == 1	Number(VAL) == 2	Number(VAL) >= 3 && Number(VAL) <= 7	Number(VAL) >= 8
Results +				
Segment	single	couple	family	large_group

- ⚡ General purpose JavaScript execution node using Node 10
- ⚡ Each node executes a JavaScript function
- ⚡ Parameters can be passed to functions from schema or public variables
- ⚡ Modifications to parameters will make changes in the input variables
- ⚡ JavaScript nodes can return objects to schema locations
- ⚡ Some useful packages are included:
 - ⚡ UUID – generate unique identifiers
 - ⚡ Moment – date handling
 - ⚡ lodash – common data structure handling
 - ⚡ ua-parser – for decoding User Agent strings
 - ⚡ crypto – for hashing or encrypting

JavaScript Node Example - Convert Fahrenheit to Celsius



- ⏪ Takes a single argument – the temperature in Fahrenheit
- ⏪ Returns the temperature in Celsius

```
JavaScript Editor ⓘ  
  
+ Add Argument  
function FtoC (fahrenheit) {  
  1 | return (fahrenheit-32)*(5/9)  
}
```

Edit 'FtoC'



↔ Replace Node

📁 Open in Script Editor

Script Return Value:

(schema)/temperature/celsius



fahrenheit:

(schema)/temperature/fahrenheit



JavaScript Node Example – Calculate Days on Twitter

JS

- Use the moment library to calculate the number of days between a day in the past and today
- Takes a single argument – the date in the past
- Returns the number of days since that date – should be a positive integer

JavaScript Editor 

 Add Argument

```
function daysOnTwitter (created_at) {
```

```
  1 | const moment = require('moment');  
  2 |  
  3 | return moment().diff(moment(created_at), "days");
```

```
}
```

Edit 'daysOnTwitter'



 Replace Node

 Open in Script Editor

Script Return Value:

(schema)/user_details/creation_time



created_at:

(schema)/tweet/user/created_at



R Script Example – Convert days to years

- ⏪ Each node executes a R Script function
- ⏪ Parameters can be passed to functions from schema or public variables
- ⏪ The return value of the function is the value of the last expression in the function
- ⏪ All arguments are passed as strings, so type conversions are necessary before manipulation, numeric values for example.
- ⏪ R Script nodes can return objects to schema locations

```
R Script Editor ⓘ  
+ Add Argument  
daysToYears <- function(days) {  
  1 #divide the time value by 365 to convert days to years  
  2  
  3 c <- as.numeric(days)/365  
}
```

Edit 'daysToYears'



↔ Replace Node

📄 Open in Rscript Editor

R Model Return Value:

(schema)/days_to_years/years














days:




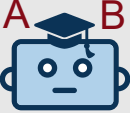





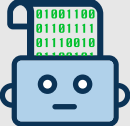

(schema)/days_to_years/days



Decision Rules and Logic

Decision Type	Description	Kitewheel Interfaces
 <p>CONDITIONAL RULES</p>	<p>Small number of distinctly different inputs with simple set of outcomes</p>	<p>IF</p> 
 <p>DECISION TREES</p>	<p>Large number of distinctly different inputs with simple set of outcomes</p>	 
 <p>DECISION TABLE</p>	<p>Simple overlapping criteria with a simple finite set outcomes</p>	 
 <p>COMPLEX DECISION MATRIX</p>	<p>Complex overlapping criteria with multi-dimensional outputs</p>	 

Advanced Analytics, Machine Learning and AI

Decision Type	Description	Kitewheel Interfaces
	PREDICTIVE MODELS Highly complex criteria modeled with existing data and outcomes	 
	ADAPTIVE MACHINE LEARNING Highly complex criteria and with a small set unknown or unpredictable outcomes	A/B 
	EMBEDDED AI Leverage a third-party web-service for Real-Time/Batch scoring or multivariate testing	 Amazon Machine Learning  
	COMPLEX SCRIPTING Data validation, manipulation, optimization and parameterization	JS 



Certification

Certification

- ◀ What types of decision nodes does Kitewheel have?
- ◀ What is the key difference between the IF conditional and the other nodes?
- ◀ When should you use a 2D Table?
- ◀ When should you choose a columnar table rather than a decision tree?
- ◀ When should you use JavaScript?
- ◀ When should you not use JavaScript?
- ◀ What JavaScript libraries are supported?
- ◀ Where are the parameter values selected for a decision node?



Thank You