**App Documentation**

**About the App:**

* This app is developed for McDonald's Australia - Delivery channel.
* This app simulates multiple pricing scenarios set by the user. Prices can be set at item level for multiple scenarios.
* The price ranges are limited to the study price ranges.
* Users can generate outputs at an overall and cluster level.

**App Build:**

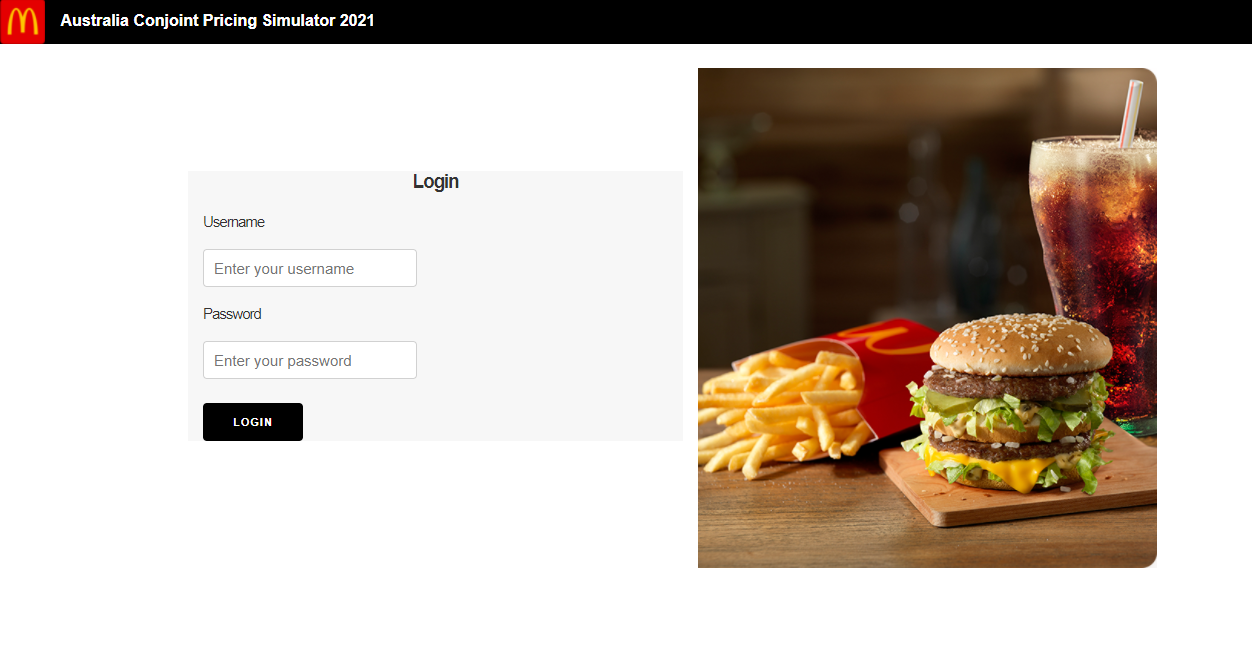
* The app is built using the flask and dash components and the plots are displayed using plotly.
* The icons used in the app are kept in the assets folder and can be modified if necessary.
* The path location to input files are kept in config file along with some standard messages that need to be displayed in the app.
* The username and password can be added to a database using “**add\_remove\_users. ipnyb.**” file.
* The log file stores the actions performed by the user along with the session ID for the particular user.
* The elasticity and about page have static inputs to be displayed and location of the files can be found in config file.

**Calibration:**

* On simulating the prices, the model gives the simulated demand at product level.
* This simulated demand needs to be calibrated to match the market demand.
* The base price used for calibration is from **May 2021** *Market data*, **March 2021** *Market data* (Chicken Salad, Chicken Salad Meal Large and McFamily Box)and **March 2021** *Transaction data* (Family McValue Box, Happy Meal (Cheeseburger, 3 McNuggets, Chicken Snack Wrap), Happy Meal McNuggets – 6Pack, Hot Cakes with Butter and Syrup Meal Medium, Late Night Bundle, McNuggets Meal 10 Pack Large, McNuggets Meal 10 Pack Medium, McNuggets Meal 20 Pack Large, McNuggets Meal 6 Pack Large, Meal for 2)
* For *Family McFavourites Box* the medium drinks price is subtracted from the provided base price to arrive at the final base price which lies within the conjoint test range.
* The units of products for actual market are taken for the period of 52 weeks starting from 2nd week of May 2020.
* Compute the market share of each product (units of one product / sum of units of all product).
* Adjust the ASC’s from model such that simulated unit shares are equal to actual market shares and update the ASC values.

**User Guide:**

**Log in:**



* On opening the app user is prompted to type in his credentials (username and password)
* Upon proper authorization the user is logged inside the app
* Unique session ID is created once the user has logged in and all actions performed all logged in a file

g

**Scenario Planner:**

h

b

i

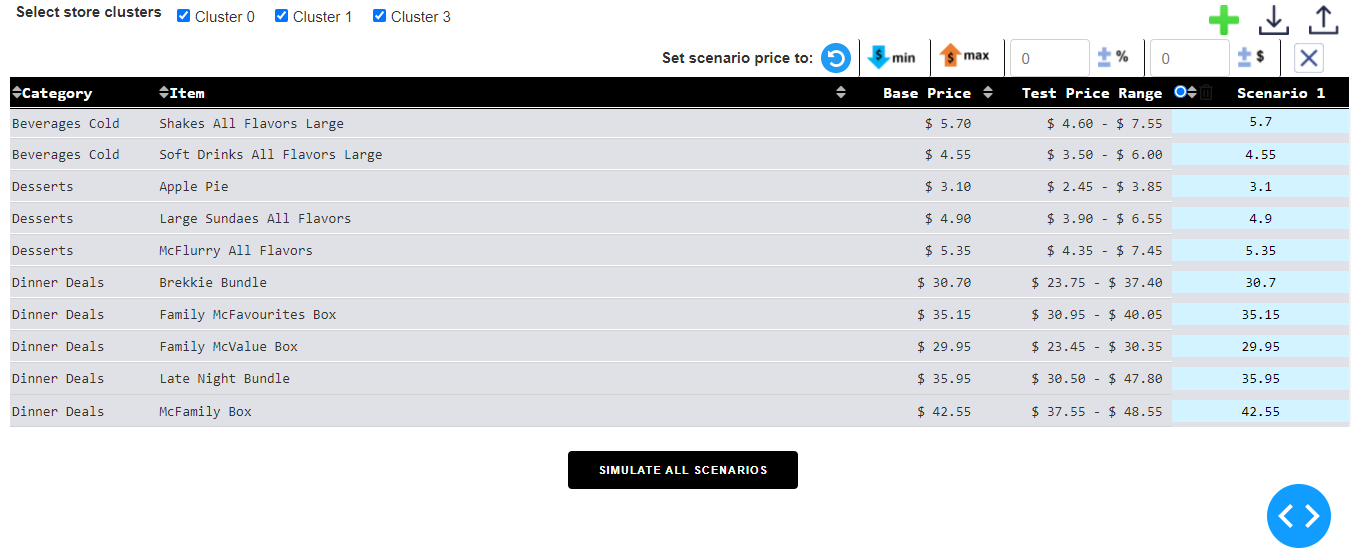
f

e

d

c

a



j



* The Scenario Planner tab of the app consists of default table with product name, category, base price, test price range and the various scenarios that needs to be run (At max only 5 scenarios can be run)
* By default, one new scenario is created with base prices.
* Users can select one or more clusters for this simulation **(a).**
* Users can reset the price of a scenario to the base price of its corresponding products **(b).**
* Users can reset the price of a scenario to the minimum tested price range of its corresponding productsusing “Min” button**(c).**
* Users can reset the price of a scenario to the minimum tested price range of its corresponding productsusing “Max” button **(d).**
* Users can change the price of all products in a scenario by certain amount or certain percentage value **(e and f).**
* Users can add more scenarios using the "add scenario" button in the menu bar **(g)**.
* Users can download a scenario of interest using the “download scenario” button in menu bar **(h).**
* Users can upload a scenario using the “upload scenario” button in menu bar **(i).** Scenarios can be uploaded in the csv/excel format.
* If uploaded scenario file does not match the format, the app throws out an error and shows the format in which the user must upload.
* There are validations to check if price of all products is within the test price range and follow the price ladder rules set by the market**.**
* If price entered for a product does not fall within the test price range of the product the price is automatically set to closest limit.
* To view the results, click on "Simulate All Scenarios" button **(j).** Generating results might take some time depending on the number of scenarios.

**Results - Cluster comparison:**

Select the scenario

Toggle between chart and table view

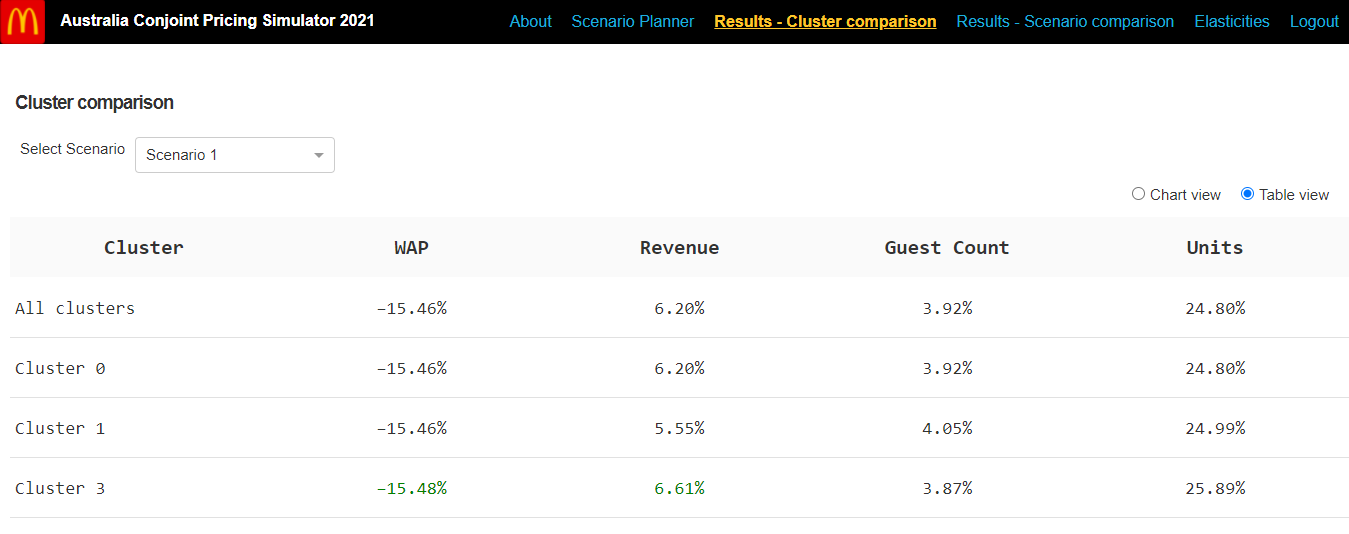


Table View

Select the scenario

Toggle between chart and table view

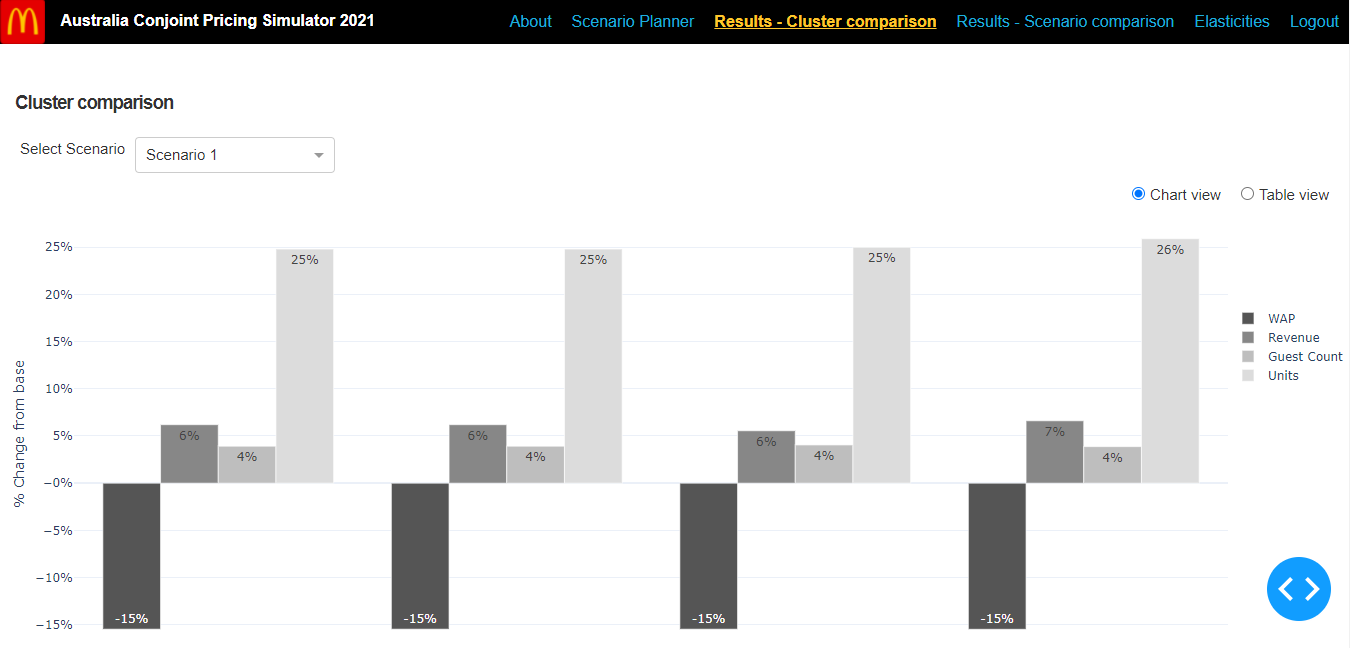


Chart view

* After Running the simulations, the user is taken to the cluster comparison results tab where he could look at the various output metrics (WAP/Revenue/Guest Count/Units) by cluster or overall level in a table or chart view.
* The chart view gives the percentage increase/decrease of the output metrics from the base across clusters in a bar chart fashion.
* The results could be downloaded in a excel format

**Results - Scenario comparison:**

Select the cluster

Toggle between chart and table

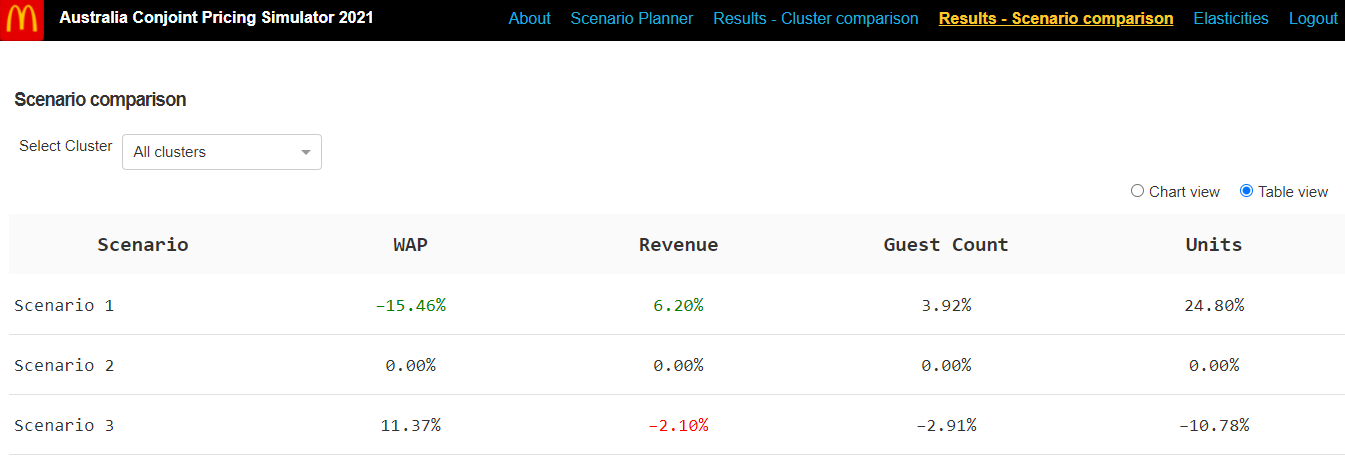


Table View

Select the cluster

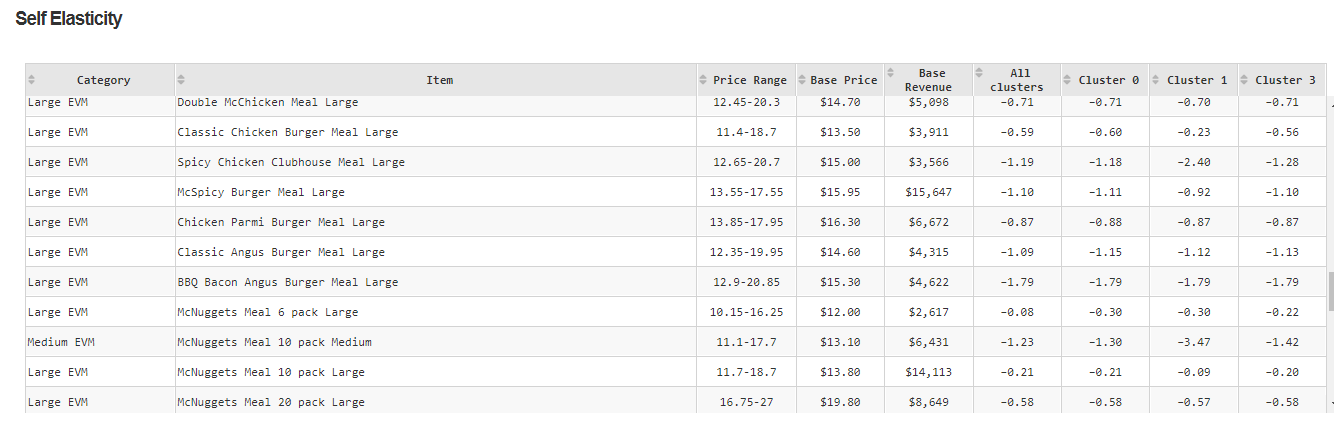
Toggle between chart and table



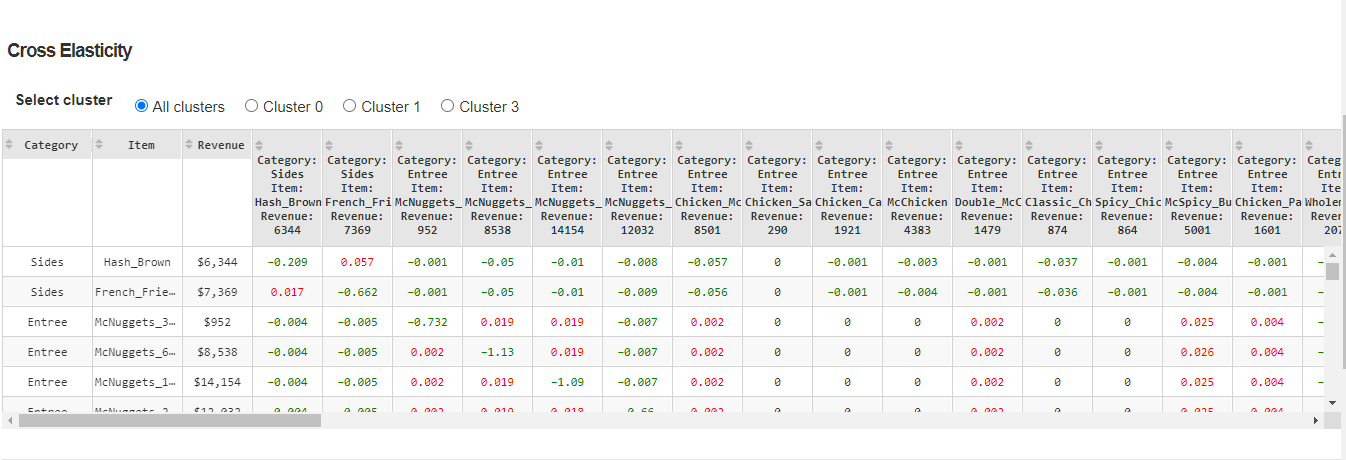
Chart View

* Scenario comparison tab compares the results across multiple scenarios the user has entered with chart and table views.
* The results could be downloaded in a excel format.

**Elasticities:**



Self-elasticity



Select the cluster

Cross elasticity

Select the cluster

Select the product to visualize

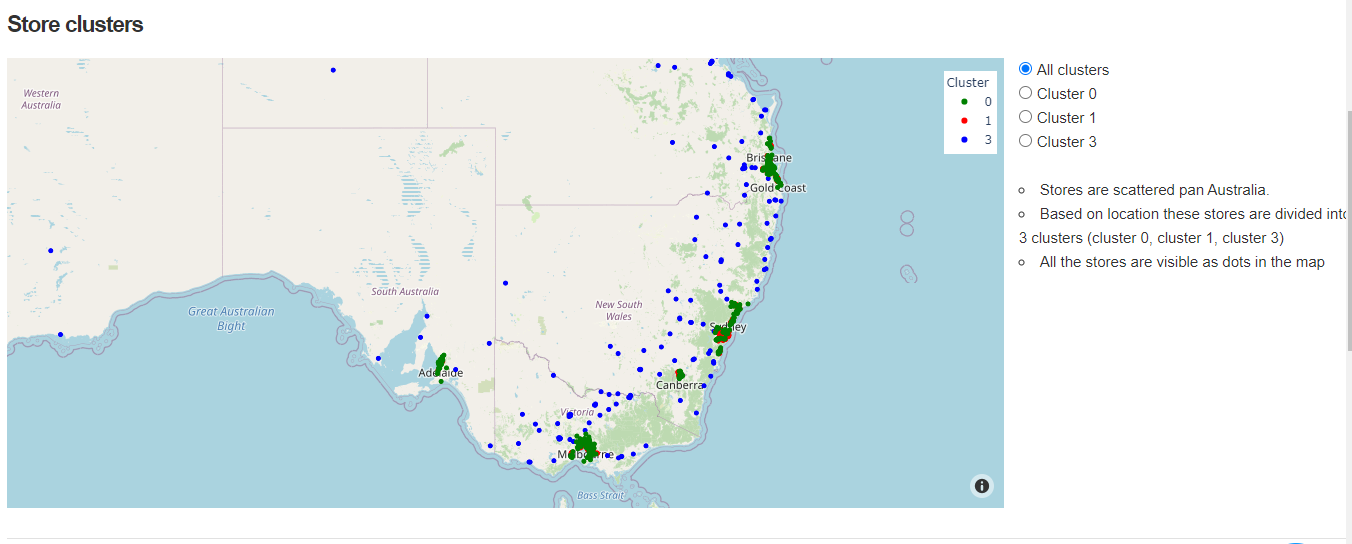


Elasticity charts

* The self and cross elasticities of the products at overall and cluster level could be found on the elasticities tab.
* Self-elasticity is the percentage change in demand of the product divided by percentage change in price of that product.
* Cross elasticity is the percentage change in demand of a product divided by the percentage change in price of some other product.
* The self and cross elasticities tab are static tables and are placed in a specific location which is mentioned in the config file.
* Elasticity analysis are line charts with price points in the x axis and some output metrics in y axis and help visualize how a product responds on moving across price points

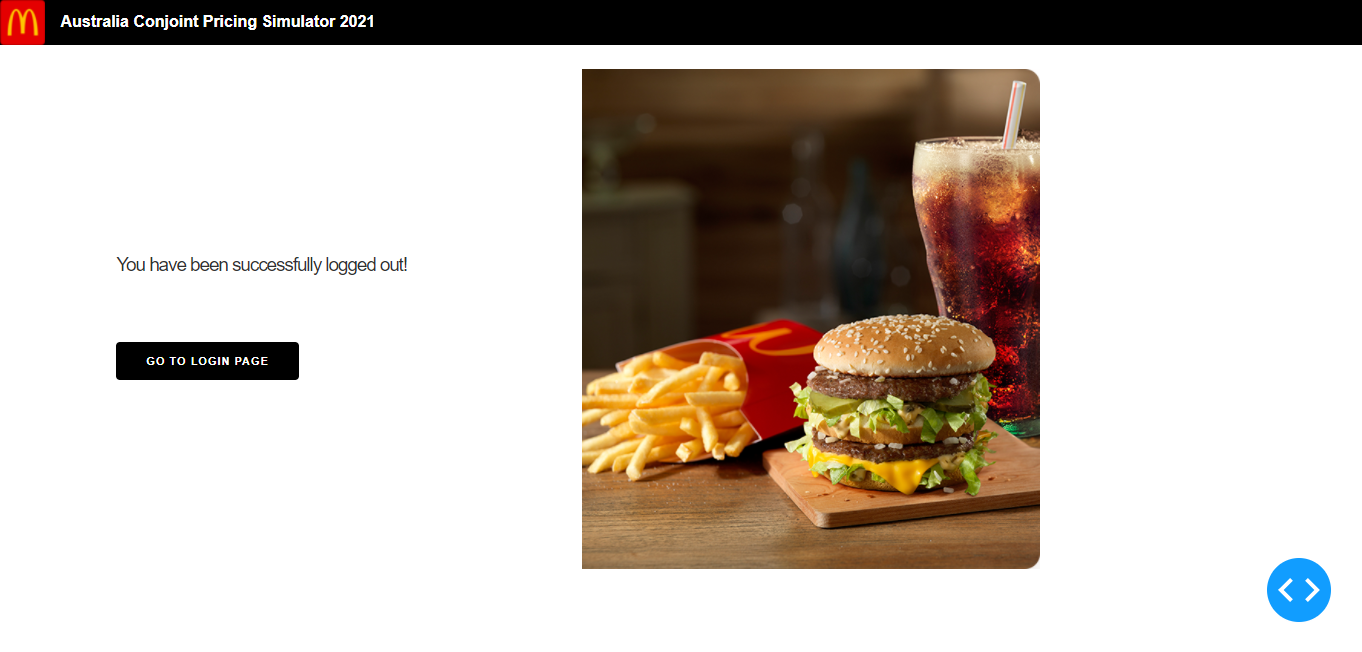
**About:**

Select cluster



* The about tab consists of a geographical map with latitude and longitude of different stores which are scattered pan Australia and divided into clusters based on location which are segregated by colour.
* It also consists of survey design and modelling methodology.
* It also gives the information about the inputs taken for simulator and the results displayed in the results tab.

**Log out:**

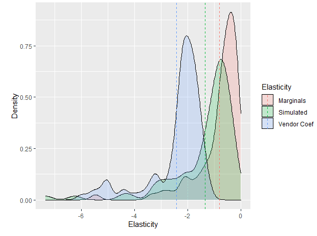


* Once the user is done with his session he can log out.

**Elasticity Comparison:**

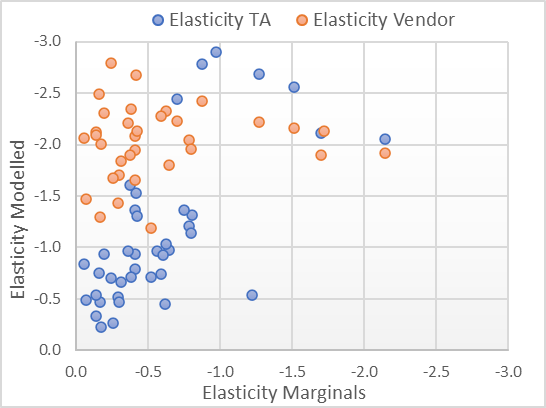
**Tiger vs Vendor vs Marginals:**

* Comparing the tiger and vendor elasticities with the elasticity from marginals, the tiger model seems to fit better as seen from below graph. The density plot below shows that the tiger elasticity seems to have more overlapping region with marginals rather than vendor



Density Plots

* Comparing the scatter plots of Tiger and vendor, more points of tiger seem to be close to the diagonal which implies that tiger model is better aligned with underlying choice data. Also the tiger model ahs a better rank correlation with marginals than the vendor.



Scatter Plots

* For more detailed comparison of elasticities please refer the following link:

<https://drive.google.com/file/d/15tWI4qEV9_qKNumJk0Ph3G-6XbDmaae6/view?usp=sharing>

**Tiger vs Vendor vs POS:**

* The POS elasticities are estimated at an item x cluster level using pooled regression model for 29 products and 75 stores (25 stores from each cluster).
* Fixed price elasticity estimated across all stores in the cluster.
* log(units) = ƒ (log(price), seasonality, trend, holidays, covid features)
* The selected items accounted for 58% of 2020 sales and the selected stores contributed to 19% of 2020 sales.
* Products which had negative elasticity in all clusters and at an overall level were chosen for comparison (8 in total).
* For comparison only considering cluster 0 and cluster 1 since cluster 3 has very low sales and units’ value across 4 years.
* MAPE error metric of Tiger and POS (45%) are less when compared with Vendor and POS (65%) across all clusters.
* In terms of magnitude, number of items in line with POS elasticity are higher for Tiger modelled conjoint compared to Vendor modelled conjoint across all clusters as can be seen from the crosstab below.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Overall** | | **Overall (0,1)** | | **Cluster - 0** | | **Cluster - 1** | | **Cluster - 3** | |
|  | **Tiger** | **Vendor** | **Tiger** | **Vendor** | **Tiger** | **Vendor** | **Tiger** | **Vendor** | **Tiger** | **Vendor** |
| **#Items on diagonal** | 4 | 2 | 4 | 2 | 4 | 2 | 3 | 1 | 4 | 2 |
| **#Items on diagonal +-1** | 6 | 7 | 6 | 7 | 6 | 6 | 5 | 6 | 6 | 5 |
| **Total** | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

Crosstab

* For generating the cross tab above elasticities are divided into bins and count the number of products falling in the bins. If more products fall in same bin as POS the better that model is aligning with POS magnitudes.
* For more detailed comparison of elasticities please refer the following link: <https://drive.google.com/file/d/1Q4CRiyfozGAC8KjfV-x6IASjOhnbPZIH/view?usp=sharing>