In this video, we will cover the key features available to the user within the Modelling module. The modelling module is where the user builds the marketing mix models.

A key aspect of the Demand Drivers platform is that it has been designed as a low-code, no-code platform. This is to eliminate any dependence on platform users needing to be proficient in analytics programming language syntax such as SAS, R, Python etc. The platform, in general, and the modelling module, in particular, utilizes a simple point and click interface. This helps to reduce the learning curve and accelerates platform adoption for business users. You do not need data scientists to build MM models in Demand Drivers.

After iterating through the input and review modules, the user can sign off on the data quality and may also create a few preliminary hypotheses. They might be able to identify potentially important predictor variables.

Let us take a few minutes to review some basic MMM concepts including an overview of the statistical regression model functional form and MMM data prerequisites

Read from Slide 1 and 2

Some of these prerequisites are that the data must be a time series. Time Series means data should have different values by day or week or month. No single row in the data should be duplicate (unique in the dimension\*time series).

Each measure selected in the model should have variability.

The number of independent measures must be less than the number of observations in the data.

It is recommended to have at least five times more observations than the number of measures

The minimum number of observations to run a model is greater than 30.

From the statistical regression model functional form perspective, the objective of a regression model is to essentially draw a straight line that passes through nearly all the actual values of the KPI.

The most basic functional form of regression is the equation of a straight line: y is equal to mx plus c.

The mathematical form is KPI is equal to intercept plus sum of coefficients into the independent measure values plus residuals, where the KPI is the predicted value of KPI being analyzed, say revenues. Intercept is the constant term, and the independent measure is the raw or transformed value of the predictor measures used in the model.

Examples of independent measures are macroeconomic factors, digital impressions, promo codes, etc. The coefficient is the rate of change of the KPI relative to the change in the independent measure, also referred to as elasticity given the model functional form, and residuals is the model error term, that is, the difference between actual and predicted values.

The example on the screen represents a typical functional form for a ride-share MMM, where the dependent KPI being modelled is First Time Users of the ride-share service. Typical predictor variables for this business are Surge Price, Holiday events, TV GRPs, Referral Bonuses, Digital Display Impressions, Precipitation etc.

Similarly, for Gaming clients, typical predictor variables could be Digital Impressions across Search and Social channels, Game Updates, Competitor activity, Holiday events, etc.

Every industry vertical has its own set of typical predictor variables.

The first step in the modelling process is for the user to define the model parameters. This includes selecting the model dependent variable. This drop-down is fed from the dependent variable box in the data classification feature in the input module.

The user must then define the model duration. By default, it is the entire period for which data is loaded into the platform. However, the user may select a more recent period if they prefer. If the user wishes to split the dataset into separate training and validation samples, they can check the Holdout option and pick a subset duration as the holdout period.

Next, the user must choose the Model Type. This decision is largely driven by the underlying characteristics of the data. For instance, if the dataset is single dimensional, the user can only choose to run an Un-pooled model.

However, if the dataset is multi-dimensional, and the objective is to run a single aggregate model, the user may choose to run a Pooled model. With a Pooled model, the model coefficients for a predictor variable remain the same across all dimension values, while the Intercept value is different for each dimension value. It increases the # of observations used to build the model by stacking up observations. Pooled model controls for differences (high and low) between dimensions

If the user has reason to believe that the elasticities for a given measure may vary by dimension value, then the user may choose to run a Mixed Effects model. Here, the user can choose which measures to run mixed effects on. The coefficients for those drivers will be generated by dimension value and at the total level. Running a Mixed Effects model is recommended when results are expected at the aggregated level + individual dimension level. An important pre-requisite to run Mixed Effects model is to **“Mean Center”** the KPI

Another option with multi-dimensional datasets is to run an Un-pooled mean-centred model, if we find that the dependent KPI needs to be normalized to avoid any skewness in the data due to high variation in the proportion of the dependent KPI across different dimension values. Mean-centering **removes** dimension level differences

Finally, the user must choose the model form. This selection refers to the underlying modelling algorithm that will be used to build the marketing mix models. Demand Drivers supports both the additive and multiplicative model forms. The Additive model form is a proprietary machine learning equivalent of the classical Ordinary Least Squares regression model form, called Quadratic Programming. The multiplicative model form is the classical log normal form. The choice of model form is once again driven by the characteristics of the underlying data and the modelling objective.

The modelling engine for Demand Drivers is written in the R programming language.

For instance, if the underlying data exhibits inherent seasonality patterns, it is recommended to use a multiplicative model form. Multiplicative models also intrinsically handle synergy effects between the incremental variables, but they do not explicitly indicate the decompositions attributed to synergy.

Additive model forms are preferred in cases where it is important to explain how we arrived at the final model. In the case of additive models, interaction terms or synergy variables must be explicitly introduced as predictor variables using the feature engineering capabilities (Create New Measures) in the Input module.

As you can see on the screen, we use the example of Average Temperature to illustrate how the user should choose between an Additive or Multiplicative model form

The blue bars in the primary y axis indicate the temperature in degrees Celsius for each month, where the secondary y axis indicates the percentage change in the first-time users that is caused by a single degree increase in temperature. In the additive model, the impact of one degree change is constant across all the months irrespective of the season. However, in the multiplicative model, the impact of one degree temperature is higher during summer and lower during winter.

In our demo example, we have just the single dimension with only one unique value. Therefore, we choose to run an Un-pooled, un-mean-centred Additive model.

Setting the model parameters is a one-time exercise and need not be repeated for each modelling iteration.

Once the user has defined the model parameters, they can do one of the following:

1. Check Correlations between predictor variables and the dependent KPI or check for multi-collinearity between various predictor variables
2. Create new predictor variables using sophisticated in-built marketing mix transforms such as Ad-stock, Gamma, Negative Exponential etc.
3. Build model iterations using all or a subset of the predictor variables that were eitheruploaded or created in the platform.

We start with the Correlation feature.

Once the user re-directs to the Correlation screen, they can select multiple combinations of groups using the 2 group-selection drop-down lists. They user can also use the date picker feature to define the period of interest for which they wish to check the correlations. This should ideally match with the model duration. For multi-dimensional datasets, the user may choose to filter their results to a subset of dimension values.

By default, the correlation limit to denote high correlation between predictor variables and the dependent KPI is set at 0.3. The user may increase or reduce this limit. This limit indicates that any variable that is more than the positive value of this limit or less than the negative value of this limit is highly correlated to the dependent KPI. In the correlation table, the highly correlated variable is highlighted in green

If one of the groups selected is the dependent KPI, then the user can click the View Chart link in the correlation output table. The pop-up displays a trend of the dependent variable and the predictor variable just as in the Review module. The user may view this chart for initially uploaded variables as well as for their transformed variants. If the measure is a transformed variable, the pop-up also displays the transformation type as well as the transformation parameters used. There is an option to download the correlation table in Excel.

We will now discuss the process to create transformed measures in the modelling module.

Often, while building the data review presentation, the user can spot patterns like consistent lags between a marketing intervention and the corresponding lift in the dependent KPI. This suggests that the input driver should probably not be used directly in the model. Instead, the user must apply a suitable transformation to this input variable such that the transformed variant can better explain the variation in the dependent KPI. Demand Drivers has a rich library of the most commonly used sophisticated marketing mix transforms such as Log, Lag, Moving Average, Gamma, Carryover, Ad-stock, Exponential, Exponential Decay, S-curve etc. that are pre-configured and available for use within the platform with a simple point and click interface.

Let us take a few minutes to review some important theoretical concepts around the need for these transformations and thumb-rule guidelines on which transformations to use under what circumstances.

Marketing science literature has published several articles that provide guidance on modelling marketing variables.

For instance, a transformation needs to reflect how digital impressions actually impact revenue. In fact, any predictor variable can be transformed prior to inclusion into a model. The purpose of transformations is to more accurately reflect the relationship being modelled and also to reduce multicollinearity.

Let's take log transformations, for example. Why do we carry out log transformations?

Taking logarithms of a variable implies that effects dampen at higher levels. (impacts become less significant as the level of the variable rises.)

For example, for unemployment as a variable, revenue decreases as unemployment rises, but past a certain point, an additional 1% rise in unemployment will not have the same impact on revenue as the first 1% rise.

Log transformation is also applicable in the case of gross domestic product (GDP).

Next, we have Dummy variables.

Dummy variables are included as controls if there are specific events that affect the KPI. Some examples where dummy variables are used are holiday events or one-off activities, etc. If we know approximately when these events transpired, we can account for them in the model using a dummy variable.

Example with large discrepancies circled in red. Dummy variables are generally created with a value = one for the days or weeks during which an event took place and zero otherwise.

Ad-stock with decay or carryover. This assumes that awareness of viewers who are exposed to media activity will decline exponentially over a period with diminishing returns.

There is one parameter to consider – which is decay.

The amount of current period activity that affects the future period where the periodicity maybe daily, weekly, or monthly.

Low values of decay indicate most of the impact was in the earlier periods while higher values of decay indicate similar impact across more days or weeks.

Parameter values range from 0.1 to 0.9. A decay rate of 0.1 means 10% impact is carried forward. When an ad influences people to make the actual purchase sometime later, we account for this lag effect. Media metrics are ad-stocked based on a specific decay rate. In this example, we have a 70% cumulative carryover.

Finally, we have the exponential decay transformation type. It assumes that consumers that were exposed to the activity make purchases that decline exponentially within a specific period. Here, the values diminish over time following an exponential curve through the number of periods specified. There are two parameters to consider –

Decay, which is the amount of current period activity that affects the dependent KPI in the current period.

Higher values decay more quickly, with most impact realized in the earlier periods.

The lower values decay more slowly with similar impact across more periods. Periods are the number of days or weeks that the decay impact is spread across. This type of transformation is frequently used with purchase incentives like promo codes. Promotions last for a limited number of periods and have no impact after that. They have a finite validity period.

Coming back to the platform.

To use this feature, the user must first select the one of the transformation types that they wish to use. They must then select all the input measures or predictor variables that they wish to apply this transformation to.

In this example, let us select the Ad-stock transformation.

We will apply this to Major Game Update Character Update and 2 other Game Update variables.

In the Transformation Details screen, for each of these variables we need to supply the Decay factor. Valid decay factor values range between 0.1 and 0.9 depending on the half-life of the specific tactic. The user may create multiple combinations of transformation parameters for the same input variable. They can do this by clicking on the plus icon and entering the number of transformed variants they wish to create for that specific tactic.

Similarly, the user can follow the same steps for all other transformation, input variable combinations. Some transformation types, such as Gamma, may have more input parameter to provide.

For the sake of saving time during this overview training video, we have already gone ahead and created suitable transformations for several of the input variables.

In cases where the user wishes to create a large number of transformations, they can use the Export and Import Excel template feature to save time.

Please note that it is vitally important to save these transformations before navigating away from this screen. If you navigate away from this screen without saving the transformations, they will be lost.

Please note that merely saving the transformations does not create the new variable in the dataset. To do this the user must click the Generate Measures button. Clicking this button reveals which measures were successfully created and which ones had errors. The user may either edit the unsaved measures that had errors and try to generate them again or delete them. The single input transforms are auto-classified into the same group as the input variable. Custom transformed variables that use more than 1 input variable in the formulae have to be classified explicitly.

Once the transformed measure is successfully generated, the newly created transformed variable is also available to be used as an input variable for a second-stage transform. For instance, an ad-stock transformed variable can then be used to create an S-curve transformed variable.

After creating the required transformed variables, the user must now select which measures to include in the model iteration. The user must scroll through the Select Measure interface and multi-select the measures they wish to use. The measures that meet the correlation limit criteria are highlighted with a green or red asterix.

After selecting the measures, the user may either run the model iteration or apply priors for some of the measures.

In some instances, the user may have some prior empirical evidence or guidance regarding the expected elasticities or contributions expected from certain drivers. This may be obtained from prior MM models or from associated media lift studies. They user may leverage this information and supply a starting coefficient value with a reasonable standard deviation. This way, the user can arrive at the optimal model iteration a lot sooner and can also strike a balance between business validity and statistical rigor.

To save time, the user may choose to download the Priors input template and fill it out in Excel before uploading it to the platform.

The user can then run the model to view the model outputs. The platform can automatically store up to 50 model iterations with the iteration names system generated. The user can compare multiple candidate model iterations for both business and statistical fit before selecting and saving the final model. The user can save multiple models if they wish.

Again, in the interest of time, we have built a model for this example that meets both criteria.

We will now look at the model outputs that are automatically generated for each model iteration.

The user can navigate to the model they wish to evaluate via the Select Model drop-down to the top right of the modelling landing page.

The model output page has 6 tabs as follows:

1. Model Fit
2. Contribution
3. Due-to
4. Effectiveness
5. RoI
6. Response Curves

We start with the Model Fit page.

This page includes the Model fit chart which is a trend plot of the actual dependent KPI vs the predicted value and includes the distribution of residuals.

The page also lists the stat fit metrics including the R-sq, Mean Absolute Percentage Error and the Holdout MAPE. A model with an R-sq greater than 90% and a MAPE less than 5% is generally considered to be a robust model.

The model fit table below the chart includes additional statistical fit metrics at the measure level. These include the Variance Inflation Factor (VIF) that is a check for multicollinearity, p-value, which is a check for statistical significance of a model measure. We also include t-stat and standard error.

From a business validity perspective, each iteration lists the model measure coefficients. The user can do a quick sense check to confirm if the coefficient signs and magnitudes make intuitive business sense.

Next, we move to the Contributions page.

This is a key modelling output as it indicates if the estimated measure-wise contributions are in line with business expectations. For example, a simple thumb rule validation could be to check if the model contribution proportions resemble the spend share proportions for the incremental drivers in the model.

The contributions are the time-aggregated decomposition values for each driver for the periods specified in the 2 period-selection drop-downs. All the base driver contributions are aggregated and represented as a consolidated positive number or percentage.

The user may use the drop-down labelled Levels to display the contribution pies at the desired level of data classification granularity.

The user can also switch to the Decomposition tab to view a trend or area-chart depiction of how the individual measure contributions have varied on a daily, weekly or monthly basis. There is an option to export the decompositions to Excel if the user wishes to conduct any additional analyses on the granular measure-wise decompositions.

We then move to the Due-to page.

The due-to chart displays the difference in the dependent KPI across 2 aggregated periods and explains the key drivers of the growth or decline between those periods. The user can drill down to a specific driver in the data grid to see the support values for the 2 aggregated periods.

Next, we have the Effectiveness page

This table indicates the effectiveness of each tactic classified as an incremental driver of the dependent KPI. It also lists the original time-aggregated value for each incremental driver that was uploaded into the platform. This value is displayed in the column labelled Support.

The Effectiveness metric is defined as the increase in the dependent KPI for every unit increase in the underlying Support for a given incremental driver. This does not apply to drivers classified in the Base variables box in the data classification step. The user can view the Effectiveness metric for each driver across different time aggregations.

Next, we have the RoI page

This is the interface where the user uploads the spends in currency terms corresponding to the underlying support values for the incremental model measures. The user can upload these spends at different time periodicities to reflect the changes in the cost of the underlying marketing tactic across different periods in time.

For multi-dimensional datasets, the user has the flexibility to upload differential spends for different dimension values if the cost of a marketing tactic varies by dimension value.

The user must also provide the RoI parameter inputs such as Coverage Factor, Revenue Multiplier and Gross Margin, if these apply to the specific business context. The default values for these are as follows:

Coverage Factor = 1

Revenue Multiplier = 1

Gross Margin = 100%

The user can use the Excel upload feature to upload the Spends, and RoI parameters if there are multiple periods and dimension values involved.

Once these inputs are provided, the user must click the Save button to confirm the changes. Failing to do so will not commit the changes in the platform.

Once the user clicks the Save button, the RoI/Efficiency, the spends, the Cost per Point charts and the RoI table render one below the other as per the time aggregation selected.

Finally, we move to the Response Curves page.

The user may either create the response curves at the time of building the model by creating response curve transformed variables prior to running the model iteration or they may create the response curves after building the first version of the model.

In the latter case, after the user navigates to the response curve page in the model output, they first need to choose whether to create multiple response curves at the individual campaign level or a single response curve at the aggregate media tactic level.

Next, for multi-dimensional datasets, the user must specify the lowest dimension level until which response curves must be generated. For each media measure in the model, response curves will be generated at that dimension and every higher dimension defined in the dimension hierarchy in the dimensions box in the data classification screen in the Input module.

Please note that before clicking on the Generate Response Curves button the user must ensure that they have entered the mandatory Measure Properties for every input and transformed variable uploaded or created in the project. The user can navigate to the Measure Properties screen under Project Settings to carry out this task.

Once the user clicks the Generate Response Curves button, the average returns response curves for all the media tactics are displayed. This may take between 3 to 8 minutes depending on the number of curves that need to be generated. The y-axis represents the average return, while the x-axis represents the percentage spend scale in 1% increments from 0% to a maximum of 300%. The dotted vertical blue line indicates the 100% spend marker. This marker represents the actual historical spend in currency terms for each media tactic across the last one year (12 months/ 52 weeks/ 365 days depending on the type of dataset. For a project with daily dataset, if the total modelling period is less than 365 days, the entire model duration period is considered for the response curves.

When the user generates the response curves using this auto-generation process, a new model iteration is automatically generated with a system generated version number appended to the previously generated model iteration. This model iteration with the response curves is automatically published to the Reporting module and is available to be shared with the Simulation and Planning modules.

As we know, modelling is an iterative process, and a user may typically have to run several iterations before finalizing a model.

To edit any of the model inputs after running an iteration, the user needs to click the Update Model button on the Model Output screen.

The user may then proceed to pick a different set of measures or edit the priors for the measures selected in the previous iteration. If the user wishes to begin a new iteration from a clean slate, they can click the Reset Parameters button in the Model Setup screen. Clicking this button clears out all the previous selections, including all the measures and priors used. All the model parameters are also reset to the default values.

The user can run as many iterations as they wish. The platform can automatically save up to 20 model iterations, by assigning a unique system-generated id to each iteration.

From the model output screen, the user can navigate to the Model History screen by clicking on the current model’s name on the top right of the model output screen.

The Model History screen displays a list of all the automatically saved models as well as the models explicitly saved by the user. The user can compare up to 4 models simultaneously based on their R-square, and MAPE values and the measure-wise contribution percentages for each model.

For every model iteration, the user can download a comprehensive summary of all the model inputs and the model outputs, as well as the data used in the model. The data includes the original data that was initially uploaded to the project as well as the transformed data that was used in the model. The user can do this by clicking on the Model Export and Data Export links on the Model output screen.

We now move to the Reporting module. This module is a view-only module for those business users that are not provided access to the first 3 modules. These users will be able to view the same reports that we saw as part of the model outputs, but only for the models explicitly saved by the modeler user. These reports include the model fit, the contribution and decomposition charts, the Effectiveness and RoI charts and the Response Curves. Unlike the Modelling output pages, the user cannot edit or update any of the fields in the Reporting module as it is a View Only module.

With this, we conclude the Modelling and Reporting modules training video. In the next video, we will turn our attention to the Simulation module where the user can use these models to run multiple marketing budget what-if scenario simulations and algorithm-driven constrained optimizations.