Hello everyone.

In this video, we will cover the key features available to the user within the Input module. We will also touch upon file types and file formats supported. As the name indicates, this module deals with the data ingestion process.

The process begins with creating a new project. A project is defined as the workspace within which the user will load the historical data, build MM models, run marketing simulations, create forecasts, and measure actual performance vs the forecasts. A project typically pertains to a single brand-market entity.

For example, this could be a mobile game title in a specific country or a CPG brand or sub-brand within a specific country or province. When a user logs into the platform for the first time, they are assigned a default project. First, they must click on the default project name on the top-right of the screen. They then click on the New Project link. The user must then fill out the mandatory fields including an easily identifiable project name, the company that the brand or product belongs to, the product name itself. This could be a brand or sub-brand or a game title depending on the industry vertical. The model date typically refers to the most recent date for which the historical data is available. Once the user enters the mandatory fields and clicks on the Save button, they are taken to the platform landing page or the Input Summary page for the newly created project.

The user may now proceed with uploading the data file. The data cube file format supported by Demand Drivers is .xlsx. The platform supports a wide variety of datasets. From a time-granularity perspective, it supports daily, weekly, and monthly data. It also supports both single and multi-dimensional datasets. Data harmonization is currently handled outside the platform. Analytic Edge can assist the client by providing data harmonization and ETL automation services to help create the data cube in the format required to be able to ingest it into Demand Drivers. Alternatively, we can provide data templates and guidelines to assist those users that wish to upload the harmonized data cube themselves. There is a comprehensive checklist to aid the users to ensure that the file uploaded meets the platform requirements. The basic prerequisite is that the data cube must mandatorily include a column representing the periodicity of the data, one column to represent any other dimensionality associated with the data. Even for single-dimensional data, the user must include this column with a single unique value. Finally, the cube must include a column representing the dependent variable or KPI for which the model will be built. The rest of the columns will represent the predictor variables.

For this series of platform overview training videos, we have used a sample mobile gaming dataset. This data represents an individual game title for a specific country. The dependent KPI is Revenue. It is a daily dataset comprising 264 days and is single-dimensional.

Once the data has been successfully loaded into the platform, the next step is to classify the data. Data classification is a one-time exercise and need not be repeated for data restatements or while loading incremental data for additional time periods.

When the user lands on this page for the first time post successful data upload, all the measures including time periodicity, dimensionality, the dependent KPI or variable, as well as all the predictor variables appear in the leftmost box titled Measures. Using a simple click and drag interface, the user must drag each measure into the relevant predefined boxes. For instance, the Date column should be dropped into the Time box. Similarly, the dimension measure “Channel”, should be dropped into the Dimensions box. Please note that for multi-dimensional datasets, it is important to drop the dimension measures one below the other in the right order such that the dimension representing the highest level of aggregation is at the top and each successive dimension represents an increasing granularity. The measure representing the highest aggregate dimension must have only 1 unique value. The dependent measure, Revenue, is dropped into the Dependent Variable box. If there are multiple candidate dependent KPIs available in the data cube, the user should classify only one of them in the Dependent Variable box and leave the other candidate dependent KPI measures unclassified.

Once the user classifies measures into these 3 mandatory boxes, they are required to click on the Approve Data button to commit these changes. Once the user confirms the tagging of measures in these 3 boxes, they cannot edit these selections.

With any MMM activity, we typically classify all the predictor variables as either Base measures or Incremental measures. For the purposes of this demonstration, given that it is a mobile gaming data set, we have further predefined sub-groups called Baseline, Media and Non-Media. For instance, organic drivers such as own brand Game Updates or Game Events could be classified under the Non-Media sub-group within the Incremental group. Example sub groups within the BaseLine group could be Holiday Events or Competition. In this example, Competition could include Game Update events from competing game titles. The group nomenclature is not rigid and is entirely user defined. The user has the flexibility to create as many groups and sub-groups (upto 6 levels) as they deem necessary. Please note that once a group or sub-group is created, the user must deposit at least 1 measure into that group before moving on to creating a new group. The entire process of creating sub-groups and classification of measures into the groups/subgroups can also be carried out by exporting the excel template and importing to save time.

The purpose of Data Classification is to facilitate a more intuitive reporting interface for the further downstream modules. Several times, we encounter models with more than 100 predictor variables. In such cases, the model outputs are easier to consume and interpret when the variables are logically aggregated using these groups.

After classifying the uploaded input measures into the relevant boxes or groups, the user may proceed to set the properties for each uploaded measure by navigating to the Measure Properties screen under Project Settings.

As the name suggests, the purpose of this exercise is to set certain important rules or properties associated with each measure.

As we can see the screen is split into 2 sections.

The left section indicates an alphabetically sorted list of all the measures or variables in the platform when the project was created. A black asterisk or star mark to the right of the measure name indicates that the mandatory properties for that measure have not been filled.

The right section of the screen indicates the fields that need to be filled for every measure, one measure at a time. This includes mandatory properties indicated by a red asterisk mark and non-mandatory properties.

The fields from top to bottom are as follows:

1. Aggregation across time – This mandatory field indicates the time aggregation rule to be applied to the measure when aggregating the value from daily to weekly to monthly to quarterly to annual values. There are only 2 aggregation options, Sum, and Average.

Example 1 – For a daily periodicity dataset, a 1-week value for the Digital Display Impressions variable would be the 7-day sum of the daily values for Digital Display Impressions variable

Example 2 – For a weekly periodicity dataset, a 1-month value for the Price variable would be the 4 or 5-week average of the weekly values for that variable

1. Aggregation rule from lower dimension level to next highest dimension – In this example, we do not see these fields as the demo dataset is a single-dimension dataset. However, if the dataset has multiple dimensions, these fields indicate the aggregation rule of Sum or Average to be applied to the measure as we calculate aggregate values from the lowest dimension in the hierarchy to the next highest dimension. If there are N different dimension levels, the Measure Properties screen will have N-1 dimension aggregation rules to be applied **mandatorily**.
2. The next field, which is also a mandatory field is Measure Type – This field has 4 unique values to choose from that may be applied to a measure

Input – This indicates that the measure was uploaded to the platform via the file upload functionality. This type of measure is not used in a model iteration

Derived – This indicates that the measure was created in the platform by applying transformations (in-built or using formulae) to the uploaded measures or other previously created transformed measures. This type of measure is used in a model iteration

Intermediate – This indicates that the measure is an intermediate transformed variant of a multi-stage transformed measure. It is neither uploaded to the project nor is it suitable for use in a model iteration.

Both – This indicates that the uploaded measure is ready to be used as is, without any transformations, in a model iteration.

1. The next field is Expected Relationship with the Dependent Measure – This field indicates the expected sign of the model coefficient for the measure. This is used as a modelling guardrail by the user or modeller to sense-check each model iteration. If a model iteration displays a coefficient that has a sign different from the one set in this field, that coefficient is flagged to the user for further scrutiny.

As a rule, all measures classified under any group or sub-group within the Incremental box in data classification are expected to have a Positive Relationship with the DV.

The 3 values that may apply here are Positive, Negative or Free Floating. If the user is uncertain of the expected relationship, they may tag it as Free Floating.

The rest of the properties are either related to display formatting (such as Number of decimals, units, scale factor, preview unit) or serve as configuration inputs to the Planning module. We will come back to these when we get to the Planning module training videos.

After setting these properties, the user must click the Save Changes button at the bottom of the screen for each measure, else the changes will be lost. Since there will typically be several measures either uploaded or created within a project, the user may find it more convenient to use the Export and Import options to enter the properties using an Excel template. After importing the filled-in Measure Properties template we need to click the Save Changes button to confirm the changes.

We go back to the Input module landing page or the Input Summary page.

As we can see, the Input module includes a few other important functions. These include the ability to edit measure labels. As the name implies, very often, the measure naming conventions used within the databases from which we ingest the data are not necessarily business friendly. Using the Edit measure labels feature, the user can provide more intuitive names for those measures. The labels will reflect across the remainder of the platform.

The Input module has an in-built feature-engineering functionality called Create New Measures. This is where the user may create additional measures such as aggregated measures, synergy, seasonality, baseline, or trend measures as well as holiday or other key events.

The user may create these new measures by using in-built arithmetic or SQL functions using the uploaded measures as arguments.

The user can also use an in-built upward smoothing algorithm to create baseline variables. For instance, we can create a Base Price measure if we wish to use undiscounted price as a predictor variable instead of average price.

The Event Flag tab provides an interface for the user to create holiday variables or any other key events that can be used in the model.

A very important feature of the input module is the automated data quality check process. As mentioned in the introduction, Demand Drivers is set up to facilitate continuous marketing and planning effectiveness measurement. To make this happen, we constantly validate model performance with the latest incremental data as it becomes available. In certain cases, an incremental data load includes a restatement of previously loaded data. The automated QC process uses a pre-configured QC file with predefined variation thresholds for each input measure. These thresholds are typically applied by a seasoned modeler with an intimate understanding of the business and the underlying data. Using pre-set aggregation rules, the platform checks across time periods for incremental data outliers. It also checks if restated data is within the threshold limits. A QC report is generated highlighting all measures for which thresholds were exceeded. The modeler may then review the report and either approve the data upload or recommend a corrected re-upload. This feature is extremely useful in terms of eliminating data errors at the inception and significantly reduces erroneous modelling iterations, thereby greatly improving the efficiency of the modelling process as well as the accuracy of the model results.

With this, we conclude the input module. In the next video, we will focus on the Review module or the Visual Data Exploratory module.