Project: Predictive Analytics Capstone

BY AYAME GOD’SWILL CLAUDE

Complete each section. When you are ready, save your file as a PDF document and submit it here: <https://coco.udacity.com/nanodegrees/nd008/locale/en-us/versions/1.0.0/parts/7271/project>

## Task 1: Determine Store Formats for Existing Stores

1. What is the optimal number of store formats? How did you arrive at that number?
2. How many stores fall into each store format?
3. Based on the results of the clustering model, what is one way that the clusters differ from one another?
4. Please provide a Tableau visualization (saved as a Tableau Public file) that shows the location of the stores, uses color to show cluster, and size to show total sales.

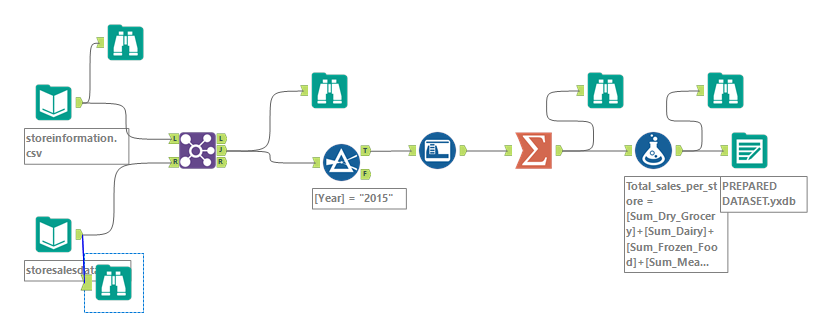
ANSWERS

The optimal number of store formats is 3.

To determine the optimal number of store formats based on sales data, I took the following steps:

* 1. **Data Preparation**
  + First, I used a JOIN tool to join the StoreSalesData.csv and StoreInformation.csv files
  + Next, I used a FILTER tool to filter only 2015 sales data which is required for the project, knocking off every other years’ sales data
  + Then I employed an AUTO FIELD tool to ensure the fields are automatically changed to the correct data types to be fed into the SUMMARIZE tool
  + Now, I used the SUMMARIZE tool to Group By ***storeID,*** ***Year, Address, City, Zip*** and then summed up to get the totals of each of the 9 category product sales for each store.
  + Next, I used a FORMULA tool to obtain the total sales for all 9 categories for each store. And then using same FORMULA tool, I obtained the percent sales for each category with respect to the total sales of all categories (calculated as: Sum category sales \* 100 / Total store sales)
  + Then I used an OUTPUT DATA tool to save my dataset to be used later

My alteryx workflow is shown below.

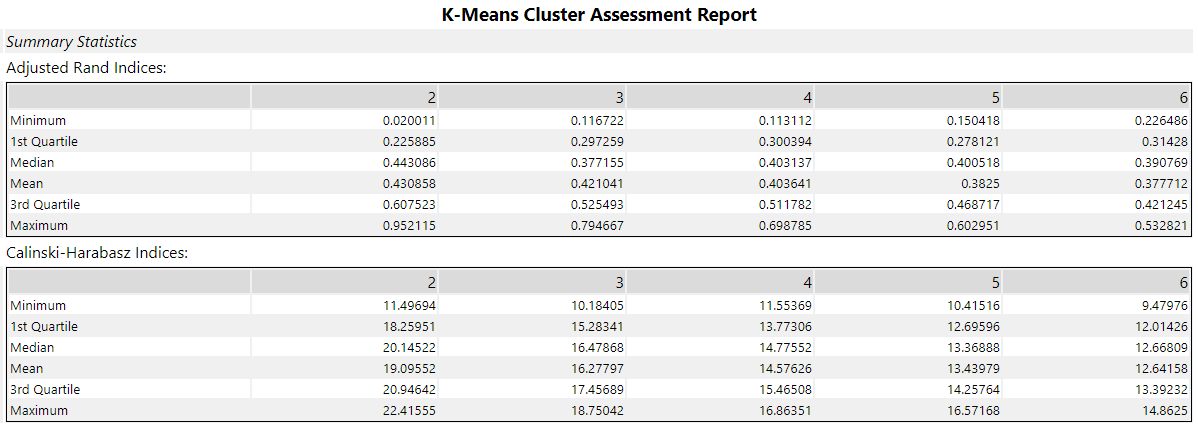


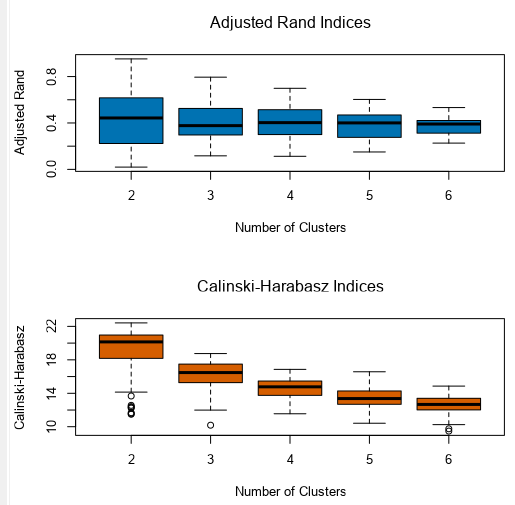
* 1. **Segmentation and clustering**

To help determine the appropriate number of clusters to use for my clustering solution, I used an INPUT DATA tool to introduce my prepared dataset into a new canvass (to reduce drag of running the workflow), and then connected the K-CENTROIDS diagnostic tool in alteryx and configured it appropriately as follows:

* As required by the project, I only used the percentage of total sales for each category as the variables in the K-CENTROIDS CLUSTER ANALYSIS tool. Thus, I unchecked every other field/variable.
* Since clustering algorithms are very sensitive to outliers, I tried eliminating this effect by standardizing the fields via checking the z-score option in the configuration panel
* I checked on K-Means for the clustering method as required for this project
* For the minimum number of clusters, I used 2 since having only 1 eliminates the need of the entire clustering process, while I used 12 for the maximum number of clusters
* Every other configuration parameter was left in their default values

Finally, in order to see the visualization, I attached a BROWSE tool and ran the workflow in order to interpret the Report.





Looking at the two indices alteryx produces – the Adjusted Rand (AR) Indices and the Calinski-Harabasz (CH) indices, I noticed that there is a box-and-whisker plot for each index value for each suggested number of clusters.

For the AR index, the higher the index the better the stability of the cluster. While for the CH index, the higher the index the better the distinctness and compactness of the clusters.

So, in general what we are looking for in both indices type is where the median/mean (indicated by the black horizontal line) of the clusters is high and the spread of the various iterations (indicated by the two extreme horizontal bars) is minimized – that is, the minimum and maximum and interquartile range are compact, suggesting fewer outliers.

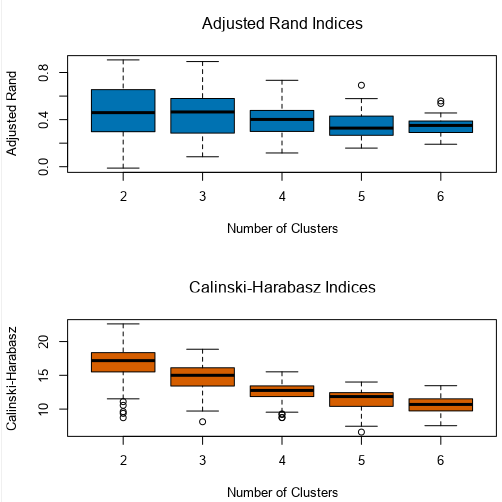
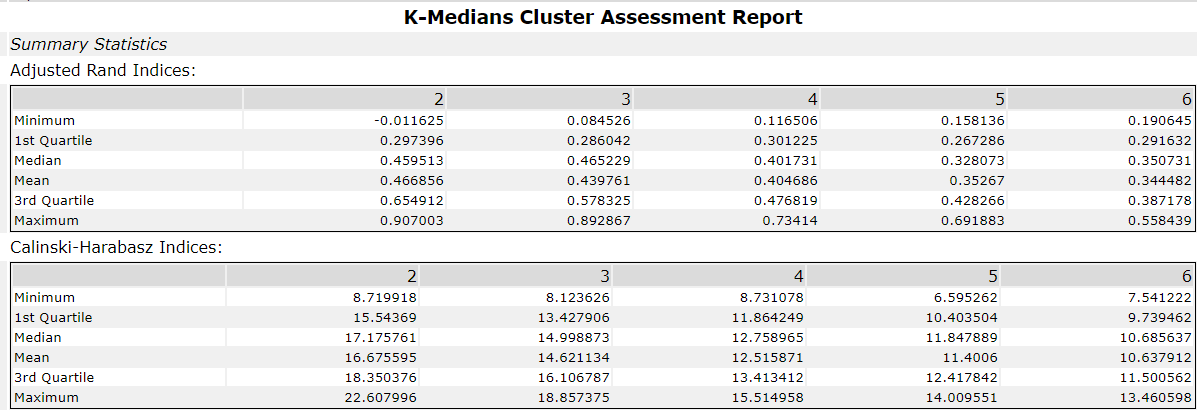
Looking at the visualization above, the AR index suggests that clusters 2 to 6 may work since they all have approximately same median value – albeit median of cluster 2 is the highest, while the spread of cluster 6 is the least.

However, the CH index suggests that 2 or 3 clusters is adequate, with 2 clusters having the highest median but 3 clusters having a better spread.

Now, these results are obviously not conclusive.

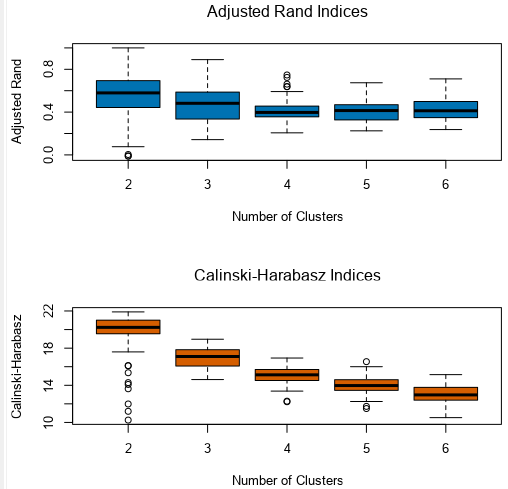
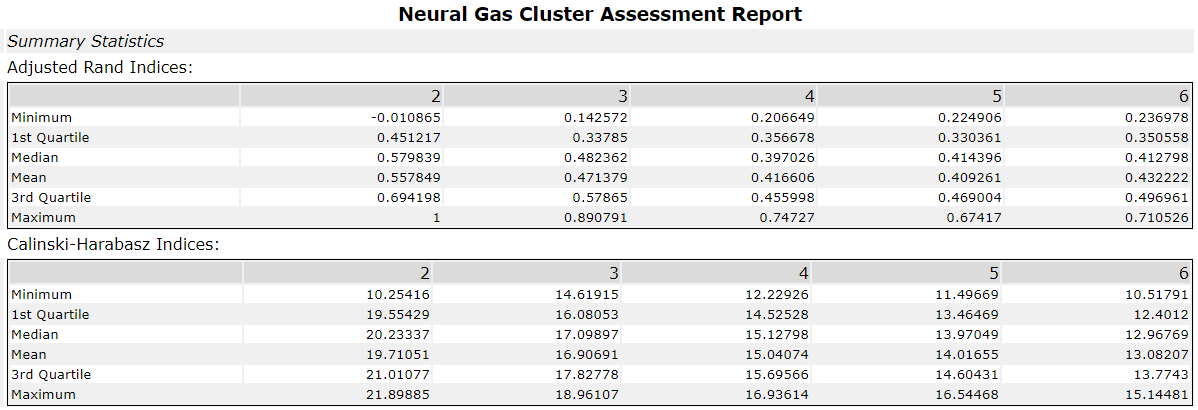
So, the next step I took is to run the same diagnostic using the two other methods available in the configuration panel of the K-CENTROIDS DIAGNOSTIC tool – the K-medians and Neural gas methods – and then compare all results to see if there is stand-out solution.

The results of the K-Medians method are shown below



As shown above, the AR indices plot suggests that the 3-cluster choice may be best since it has the highest median and its spread is relatively okay. While the CH indices plot suggests that 2-clusters choice may be best since it has the highest median, but in terms of spread, 3-clusters choice is better.

Finally, I changed the K-clustering method in the configuration panel to Neural Gas and ran the workflow. The results are as shown below.



The visualization above for AR indices plot suggests that 3-clusters choice may be best since its median and spread are fairly ok, while the CH indices plot seem to give more credence to 2-clusters, but the median and spread of the 3-clusters choice is good.

Looking critically at the results from all three methods of clustering, I decided to make use of the K-Means clustering AR and CH indices plot, as its results appeared much clearer and thus arrived at the optimal number of store formats (segments/clusters) to be 3.

My workflow is shown below.

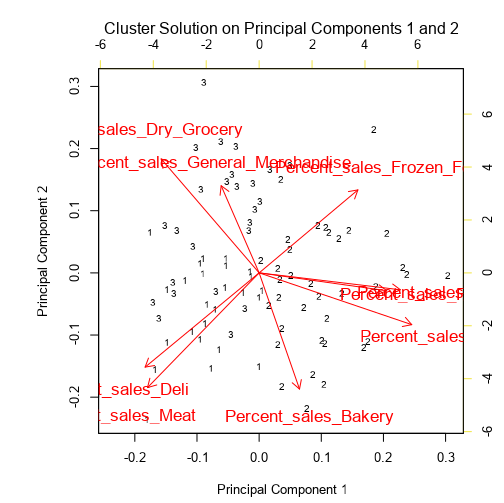
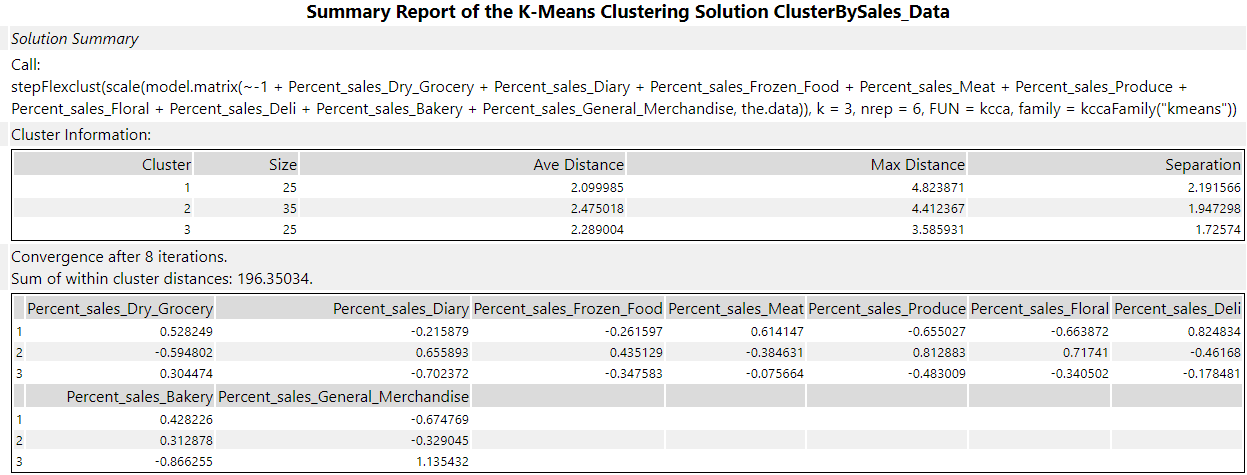


3. **Creating the cluster model**

I imported my prepared dataset into a new canvass in alteryx, and then connected the K-CLUSTER ANALYSIS tool. In the configuration panel, I gave a cluster solution name ***clusterBySales\_Data,*** and then ensured that I used exact same configuration I employed in choosing my optimal number of store formats used in the K-CENTROIDS DIAGNOSTIC tool

– which includes selecting same fields, same standardization and most importantly the K-Means clustering method.

However, I used my predetermined number of clusters as 3 and increased my number of starting seeds from 3 to 6 (as it has been noticed it helps to improve the accuracy of the overall cluster solution, especially with respect to the iterative process. Then I ran the workflow and obtained the following results from the R-output.



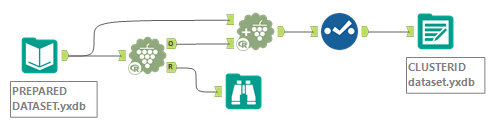
As shown in the Size column of the report above, 25, 35 and 25 stores respectively fall into clusters 1, 2 and 3. More so, the stores in cluster 1 have the least intra-cluster distance, that is, average distance from their centroid (central point) indicating compactness, followed by cluster 3 and then cluster 2, as shown in the Ave Distance column

Furthermore, the separation of the stores in cluster 1 from the other clusters (inter-cluster distance) is greatest, also affirming that the stores in cluster 1 are more compact and must have fewer outliers, as shown in the Separation column. The more separation there is the better for the model.

The 2nd chart in the summary report shows the inter-cluster comparisons for each variable. For instance, for ***Percent\_sales\_Dry\_Grocery***, cluster 1 has the highest positive value while cluster 2 has the highest negative value. This implies that the stores in cluster 1 are the greatest opposites to those in cluster 2 in terms of the variable/field ***Percent\_sales\_Dry\_Grocery.*** However, the positive and negative values do not necessarily mean that the stores in cluster 1 have a higher ***Percent\_sales\_Dry\_Grocery*** than those in cluster 2. In actuality, it could be that those in cluster 2 did better – the positive and negative values are only meant to indicate that they are most different, like the two ends of a pole or greatest disparity. Thus, the store with the greatest ***Percent\_sales\_Dry\_Grocery*** could actually have a negative value while that with the least ***Percent\_sales\_Dry\_Grocery*** could have a positive value, vice versa (which can be easily determined by validating the results).

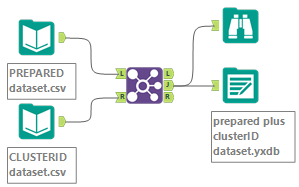
The next thing I did was to use the cluster model I created to identify objects with a cluster identifier. To do this, I added the APPEND CLUSTER tool in alteryx and connected the O-output of the K-CENTROIDS CLUSTER ANALYSIS tool to the 2nd input of the APPEND CLUSTER tool, while connecting the original dataset to the 1st input of the APPEND CLUSTER tool. Next, I added a SELECT tool to the output of the APPEND CLUSTER tool in order to output only the ***storeID*** and ***clusterID*** fields. Finally, I added an OUTPUT DATA tool to save the file to be used in combination with my raw prepared dataset for further validation. Then I ran the workflow.

My workflow is shown below



5. **Validating and applying clusters**

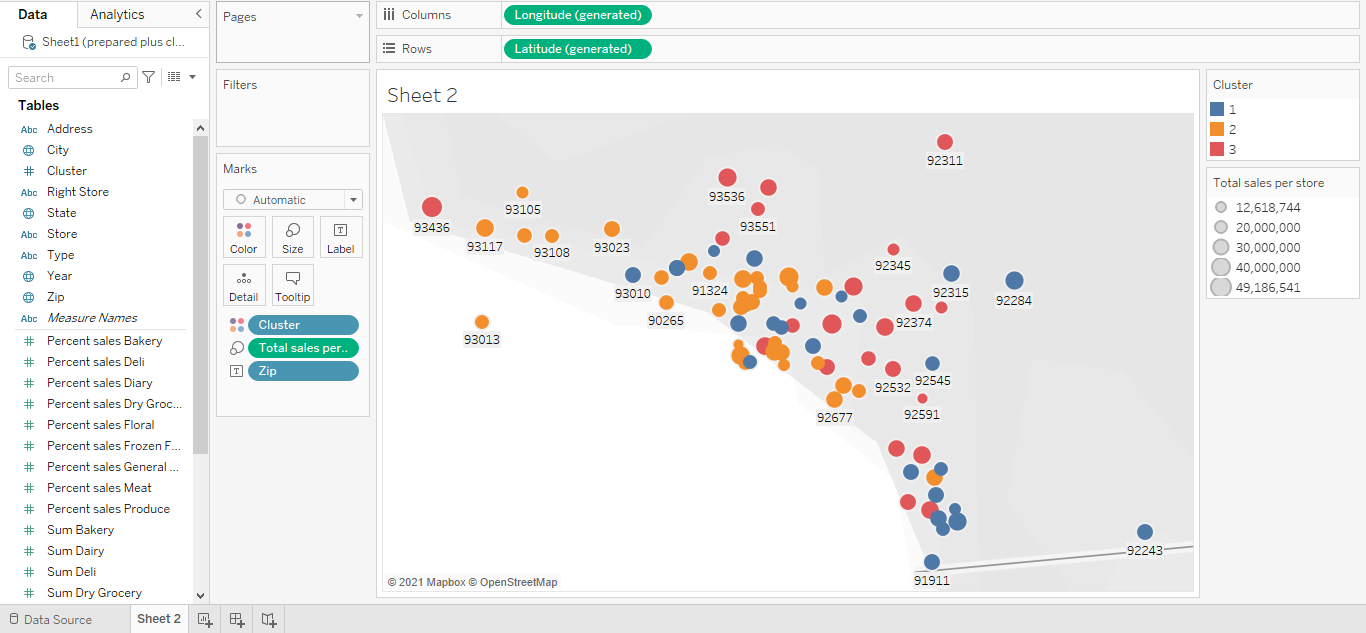
To provide some visualisations in Tableau, I used a JOIN tool in alteryx to combine both my raw PREPARED dataset and my CLUSTERID dataset and then saved as a csv so that it can be accessed by Tableau to generate some visualizations to see if my solution makes sense.



I then brought the combined dataset into Tableau.

First, I converted the ***cluster*** field to Dimension since it is actually a dimension, not a measure. And I also right-clicked on the ***year*** field and changed the Geographic Role to Area Code (US) in order to reduce the occurrences of “unknown locations. Next, I tried to see if the clusters look different when looking at them on a map. To do this, I dragged the generated Longitude and Latitude fields to the columns and rows sections respectively. Then I clicked on Analysis on the top left-hand side of the window to uncheck Aggregate Measures so that it will show the clusters separately. Finally, I dragged ***cluster, Total sales per store and Zip*** to the “Color”, “Size” and “Label” panes respectively to enhance the visualization. After this, I noticed “unknown locations” at the bottom right-hand side of the window, which I clicked on and changed the location to “United States”.

The final output produced is as shown below.



## Task 2: Formats for New Stores

1. What methodology did you use to predict the best store format for the new stores? Why did you choose that methodology? (Remember to Use a 20% validation sample with Random Seed = 3 to test differences in models.)
2. What format do each of the 10 new stores fall into? Please fill in the table below.

|  |  |
| --- | --- |
| Store Number | Segment |
| S0086 |  |
| S0087 |  |
| S0088 |  |
| S0089 |  |
| S0090 |  |
| S0091 |  |
| S0092 |  |
| S0093 |  |
| S0094 |  |
| S0095 |  |

## ANSWERS

The grocery store chain has 10 new stores opening up at the beginning of the year. The company wants to determine which store format each of the new stores should have – that is, which segment each of the new stores should be placed or clustered into. However, we don’t have sales data for these new stores yet. Hence the only option is to determine the formats/segments/clusters using each of the new stores’ demographic data – demographic and socioeconomic characteristics of the population that resides in the area around each new store – and try to find old stores in the already-created clusters that share similar demographic data characteristics so that the new stores can be clustered along with the appropriate corresponding old stores.

To begin, I used a JOIN tool in alteryx to combine my Prepared\_plus\_clusterID dataset with the given storedemographic data (containing demographic information for the new stores). Now, inside the JOIN tool, I changed the data type of all the demographic variables (predictor variables/fields) from V\_String to Double (since all the values are in decimals), while that of the ***cluster*** field (target variable) was left as V\_String (since it is actually a categorical variable).

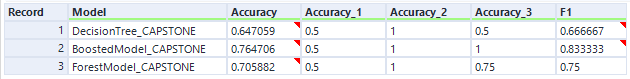
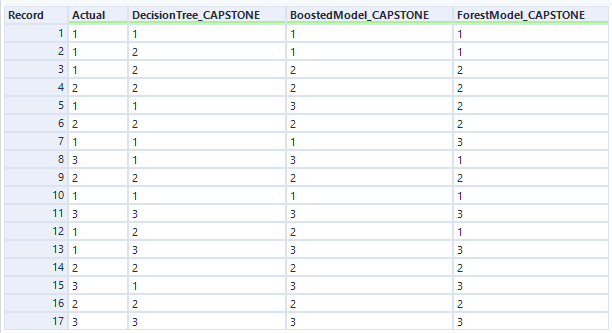
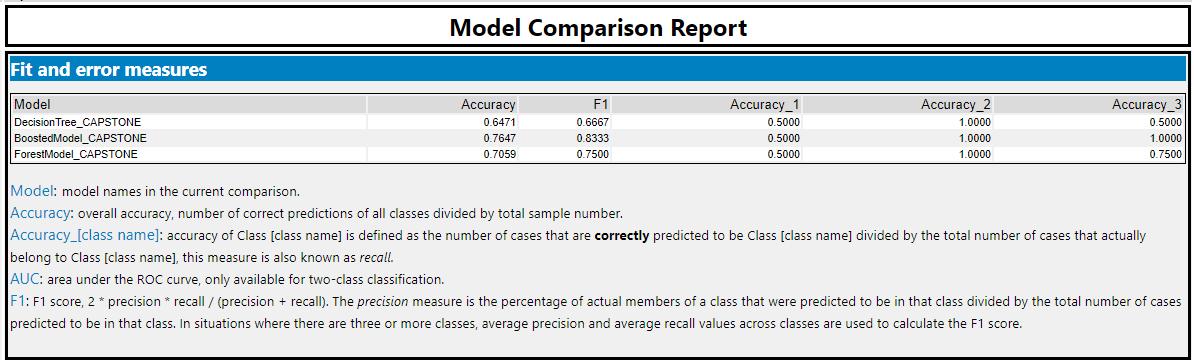
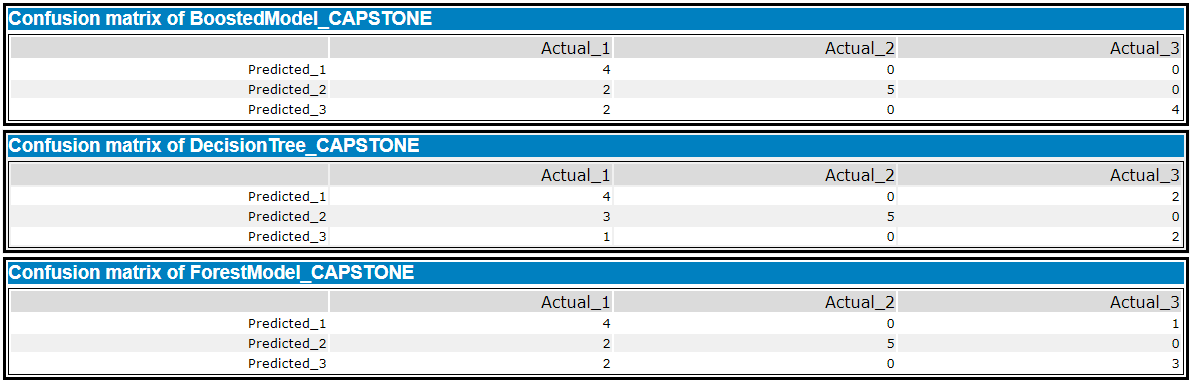
Next, I created my Estimation and Validation samples (with 80% of my entire training dataset going to Estimation and 20% reserved for Validation) and setting the Random Seed to 3 in the CREATE SAMPLES tool in Alteryx.

Then, I introduced the DECISION TREE, FOREST MODEL and BOOSTED MODEL tools and connected all three of them to the E-output of the CREATE SAMPLES tool in Alteryx.

Next, I used the UNION tool to union all three models together, and then the output of the UNION tool was connected to the M-input of the MODEL COMPARISON TOOL whilst connecting the V-output of the CREATE SAMPLES tool in Alteryx to the O-input of the MODEL COMPARISON tool, and in the configuration panel, I left “The positive class in target variable” blank since the target variable in this case is not binary (since we are dealing with 3 store formats).

Finally, I connected BROWSE tools to the 3 outputs of the MODEL COMPARISON tool and ran the workflow. The aim was to compare all three models side-by-side to ascertain which is the best model.

The results are shown below.

As can be seen in the results above, the **Boosted Model** performed the best. From the Model Comparison Report, the overall percent accuracy of the Boosted model is 0.7647 or 76.47% which is strong and the highest of all three models tested. More so, it’s accuracy in predicting store format 1, 2 and 3 is 0.5000 or 50%, 1.0000 or 100% and 1.000 or 100% respectively (which were either the best or as good as others).

Additionally, looking at the F1 column we can reach same decision. The F1 score is the weighted of precision and recall which takes both false positives and false negatives. The higher the F1 score the better the model’s predicting ability. Here again, the boosted model has the highest value of 0.8333.

Thus, the boosted model was used to choose the best store formats for the new stores.

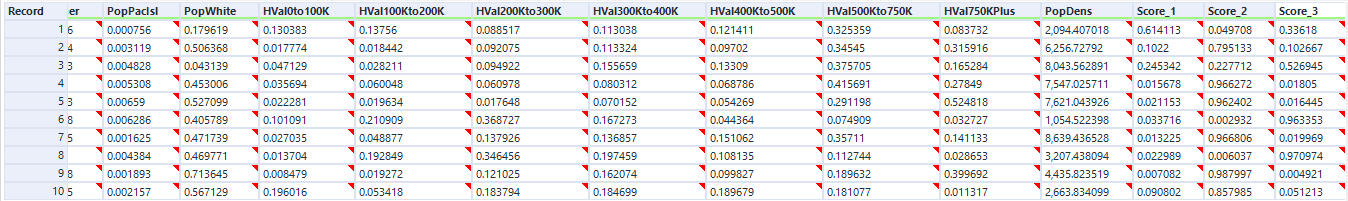
Next, the INPUT DATA TOOL was used to introduce the storedemographics dataset to be scored to the same workflow, and an AUTO FIELD tool in alteryx was added to convert the demographics variables/fields to double format while store in string format like before. Then I ran the workflow to effect the fields datatype conversions. After which I added a SELECT tool just to confirm that all the required variables were actually converted appropriately.

Then I connected a RECORD ID tool in order to create a new column called ***RecordID*** which gives the correct numbering of the records (since the numbering in EXCEL is usually one value higher than normal) that will enable me to filter the values as required and assign a unique identifier/cluster to each record/row/store.

Now, the storeinformation dataset shows that the 10 new stores which have to be scored have store IDs from S0086 to S0095. Hence, I connected a FILTER tool to filter to retain the corresponding last 10 records or stores in the storedemographics dataset.

Then I connected the T-output of the FILTER tool to the O-input of the SCORE tool whilst connecting the O-output of the BOOSTED MODEL to the M-input of the SCORE tool in alteryx.

A BROWSE tool connected to the output of the SCORE tool produced the following results



Now, it is worthy to note that each of the values under the columns ***score\_1, score\_2*** and ***score\_3*** indicates the probability of a store falling into any of cluster 1, 2 and 3 (represented here as ***score\_1, score\_2*** and ***score\_3*** respectively. The final conclusion as to each store format/cluster/segment would be determined by connecting a FORMULA tool to the score tool and using the expression:

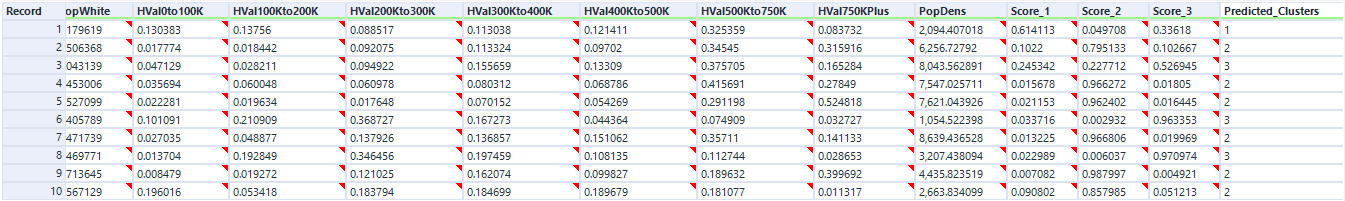
IF [Score\_1]>=[Score\_2]and [Score\_1]>=[Score\_3] THEN "1"

ELSEIF [Score\_2]>=[Score\_1] and [Score\_2]>=[Score\_3] THEN "2"

ELSE "3"

ENDIF

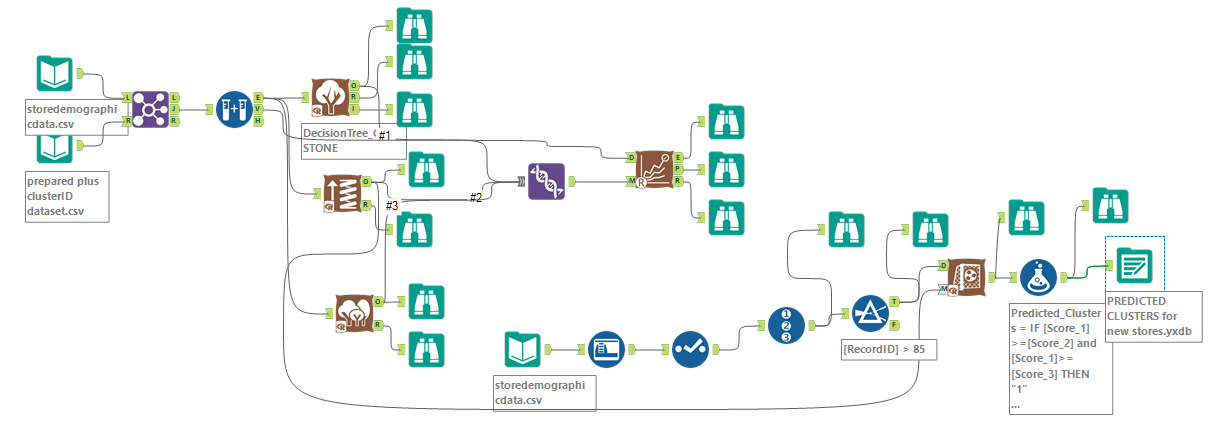
A BROWSE tool connected to the output of FORMULA tool produced the following results.



|  |  |
| --- | --- |
| Store Number | Segment |
| S0086 | 1 |
| S0087 | 2 |
| S0088 | 3 |
| S0089 | 2 |
| S0090 | 2 |
| S0091 | 3 |
| S0092 | 2 |
| S0093 | 3 |
| S0094 | 2 |
| S0095 | 2 |

Finally, I connected an OUTPUT DATA tool and ran the workflow in order to save the new stores cluster dataset to be used later.

My solution workflow is shown below.



## Task 3: Predicting Produce Sales

1. What type of ETS or ARIMA model did you use for each forecast? Use ETS(a,m,n) or ARIMA(ar, i, ma) notation. How did you come to that decision?

2. Please provide a table of your forecasts for existing and new stores. Also, provide visualization of your forecasts that includes historical data, existing stores forecasts, and new stores forecasts.

**ANSWERS**

First thing to do is to determine the type of ETS or ARIMA model to be used for each forecast – existing stores and the new stores. Finally, we will use this model to forecast sales/produce for the existing stores, and then the 10 new stores.

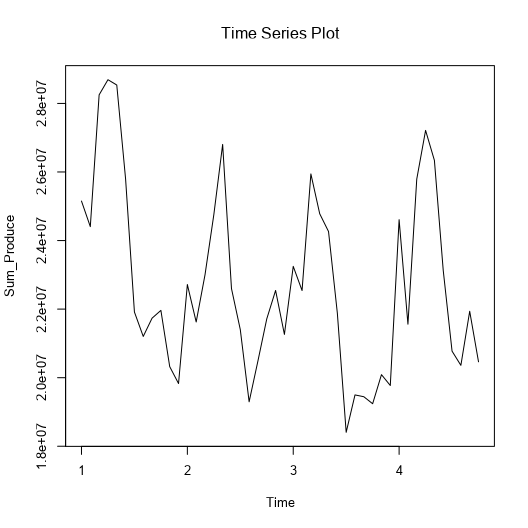
**Determining the model type**

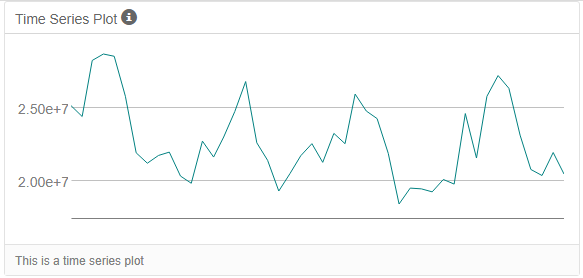
I used the INPUT DATA tool to introduce the storesalesdata into a new canvass in alteryx. Then I connected an AUTO FIELD tool to convert to appropriate data formats (since it’s a csv file), ran the workflow to effect the conversions and then connected a SELECT tool to confirm the changes.

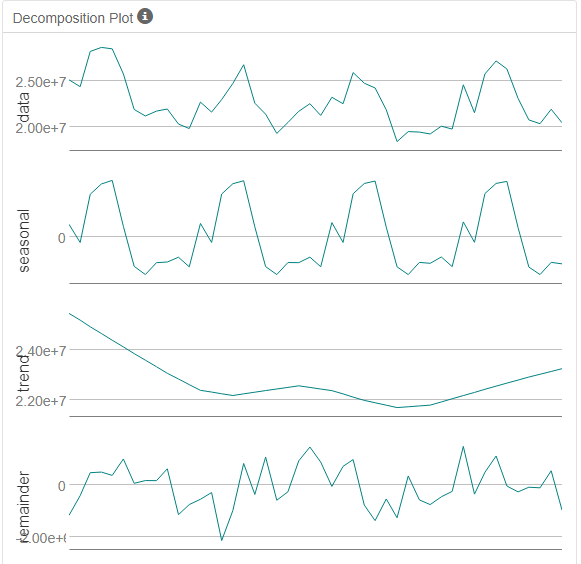
For existing stores, we are trying to get the monthly total for past stores before forecasting. I accomplished this by connecting a SUMMARIZE tool and

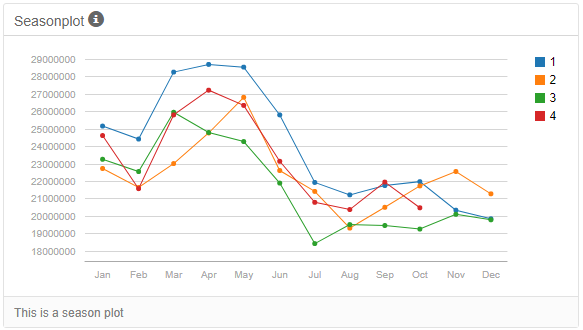
* Grouped by Year
* Grouped by Month, and then
* Sum Produce.

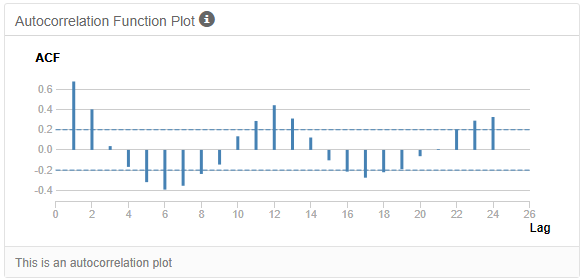
Next, to determine the fashion of how to apply my Trend, Seasonal and Error components, I connected a TS PLOT tool to the SUMMARISE tool and configured it such that ***Sum\_Produce*** is my target field, I selected Monthly as my target field frequency (since the company wants a monthly sales forecast), and Time Series Plot as my Plot type. Then I added a BROWSE tool to the outputs and ran the workflow. The results are shown below.

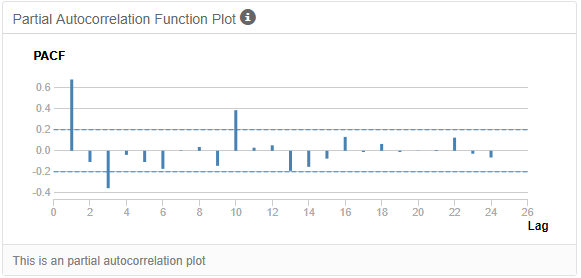












**Interpretation:** The Decomposition plot helps us determine the type of ETS model to use to build our model, while the ACF and PACF plots helps us determine the type of ARIMA model to use to build our model. The plan is: After determining the particular type for each model, we will then use a TS COMPARE tool in alteryx to compare both models to see which model is actually the best to rely on to make our forecasts both for the existing and new stores.

Now, ETS is short for Error Trends and Seasonality, describing the three key components of an ETS model.

The Decomposition Plot above shows that the Trend first goes down and then slightly up, down and up again. Hence, we cannot say there is a general Uptrend nor can we state a Downtrend. Hence there is No trend (thus we use N for the T component in ETS).

The seasonal portion of our decomposition plot helps us to confirm if the result we reached about Trend above is correct. Looking at the seasonal peaks and valleys, we can see that seasonality increases slightly along the time series. Thus, it is Multiplicative (M). This suggests applying seasonality as M in our forecast tool later.

Lastly, the Remainder or Error portion of the Decomposition plot shows change in variance as the time series moves along. This suggests we should use Error Multiplicatively (M).

Thus, we have ETS (M, N, M) model.

Now, for the ARIMA model, looking at the ACF (AutoCorrelation Function) and PACF (Partial AutoCorrelation Function) plots we see the serial correlation of the series or how correlated it is with itself.

Now, if the ACF plot shows autocorrelation decaying towards zero and the PACF plot cuts off quickly towards zero, then it is an indication that we need to include “AR” in our ARIMA model (with AR short for AutoRegressive, aka “p” standing for number of **P**revious periods). More so, if the ACF shows positive correlation (positive slope) at Lag-1, then AR terms are best. Looking at my ACF and PACF plots, none of these exists. Thus we have AR or p = 0.

However, if the ACF shows negative correlation at Lag-1, ACF cuts off shortly after a few lags, and a PACF that deceases out more gradually, then we should include an “MA” in our model (with MA short for Moving Average, aka “q”, that is, Error or Remainder, just like in ETS models). Looking at both ACF and PACF plots, this holds true here. Thus, we have MA or q = 1.

The integrated (I) component term refers to the number of times we have to difference (d) our dataset to make it stationary. If the series must be differenced to make it stationary (so we can easily forecast with it), then we say it is integrated. If we only need to take the first difference to stationarise our series then the I term will be 1. Now, since my ACF and PACF plots shows that the I or d = 1.

Hence, I arrived at ARIMA (0, 1, 1) model, in accordance with the format ARIMA(p, d, q).

***NB: ARIMA(p, d, q) is the format for non-seasonal ARIMA models, where p stands for the number of AR or previous periods, d is the degree of differencing and q is the number of moving average or error terms.***

***For seasonal ARIMA models (that is, models exhibiting seasonality), there are some extra additions, and the format is: ARIMA(p, d, q)(P, D, Q)m***

***Where (P, D, Q) stands for the seasonality values for the AR, I and MA components respectively, while m stands for the number of periods in each season (e.g. m=12 for a monthly data spanning a 1-year period)***

Next step is to build both models.

We begin with the ETS using an ETS tool. So, I connected a RECORD ID tool to the SUMMARIZE tool in order to create a new column called ***RecordID*** which gives the correct numbering of the records (since the numbering in EXCEL is usually one value higher than normal) that will enable me to filter the values as needed. Then I connected a FILTER tool to filter out the last 6 records/rows which will be used as the Holdout Sample (since project requires us to use a 6 month holdout sample for the TS Compare tool judging from the fact that we do not have that much data so using a 12 month holdout would remove too much of the data) by configuring the FILTER tool to include only values less than or equal to ***RecordID*** 40. This process has helped me prepare my dataset for forecasting.

So, I connected an ETS tool to the T-output of the FILTER tool, and in the configuration panel named the model MNM (to indicate how I am applying the ETS components), used ***Sum\_Produce*** as the target field, and selected Monthly as the target field frequency (since the company wants a monthly sales forecast). Next, I clicked on the Model Type tab to set the different component types (using the MNM format determined above). However, to ensure accuracy I checked the “Auto” options whilst selecting No for Trend dampening.

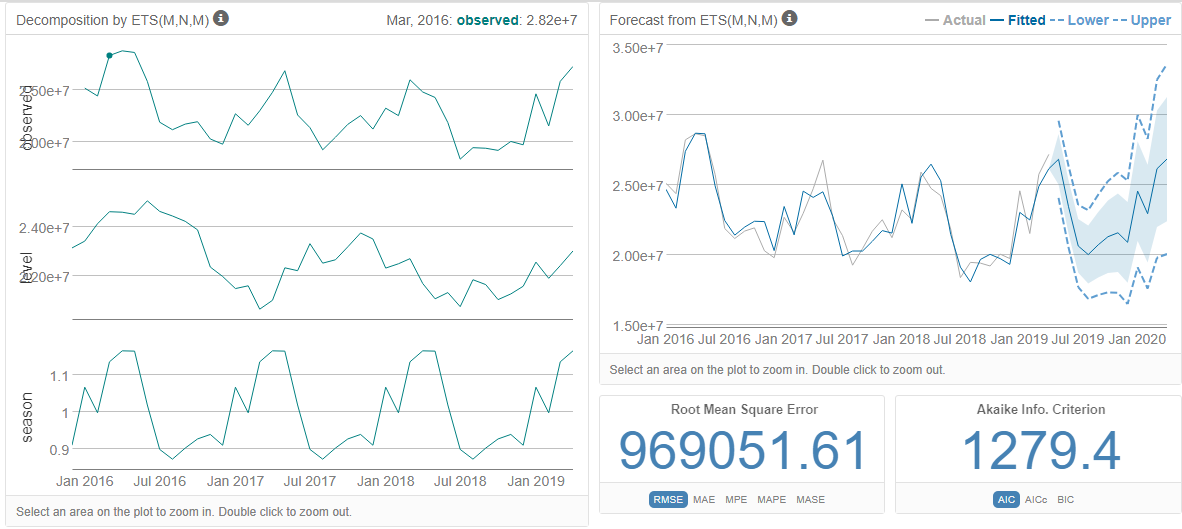
***NB: The “Auto” options under the model type tab can be used to automatically determine the best ETS format without having to use the TS PLOT tool nor going through the steps taken above. But for this project, it was required that we manually discover the best format (as it is important to get an understanding of how it works), hence the processes above.***

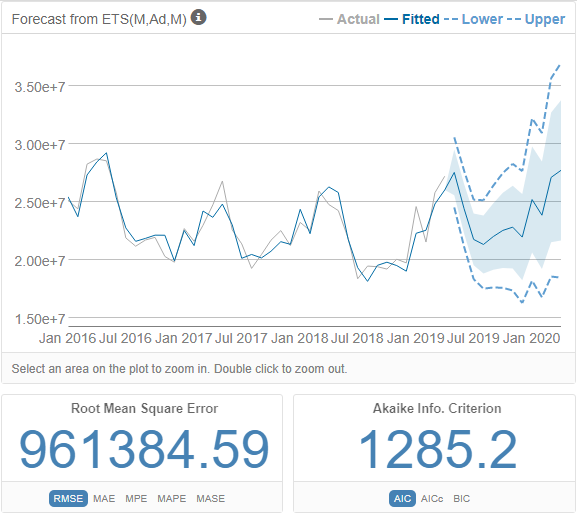
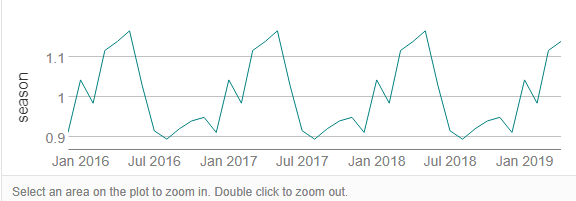
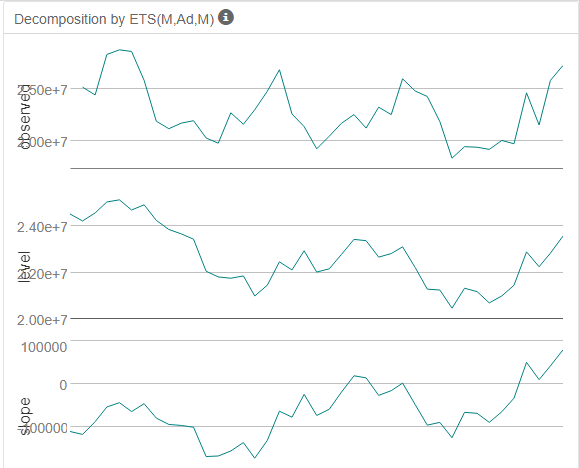
Now, under the “other options” tab, I chose 12 as the number of forecast periods (since we were asked to prepare a monthly forecast for produce sales for the full year of 2016, that is 12 months, for both existing and new stores.), and then set the “series starting period” to 2016 period 1 (that is, 1st month).

Now, to test if Trend dampening would have a positive effect on our model, I copied and pasted the ETS tool unto the canvass, connected it to the T-ouput of the FILTER tool like before and ensured same configuration settings, except checking Yes for the “Trend dampening” option plus adding the word Dampened to the model name in order to help differentiate it from the first.

Finally, I added BROWSE tools to the outputs of both models, and ran the workflow.

The R-output of the ETS models showed me some important information, such as in-sample air measures, information criteria and the selected alpha value.





Now, the better model is usually chosen as the one with the lower Root Mean Square Error (RSME) value and a lower Akaike Information Criterion (AIC) value.

Looking at the results shown above, we can see that while the ETS model with Trend Dampening, ETS(M, Ad, M), has a lower RSME, its AIC value is higher.

Thus, after weighing all factors, I decided to go for ETS(M, N, M).

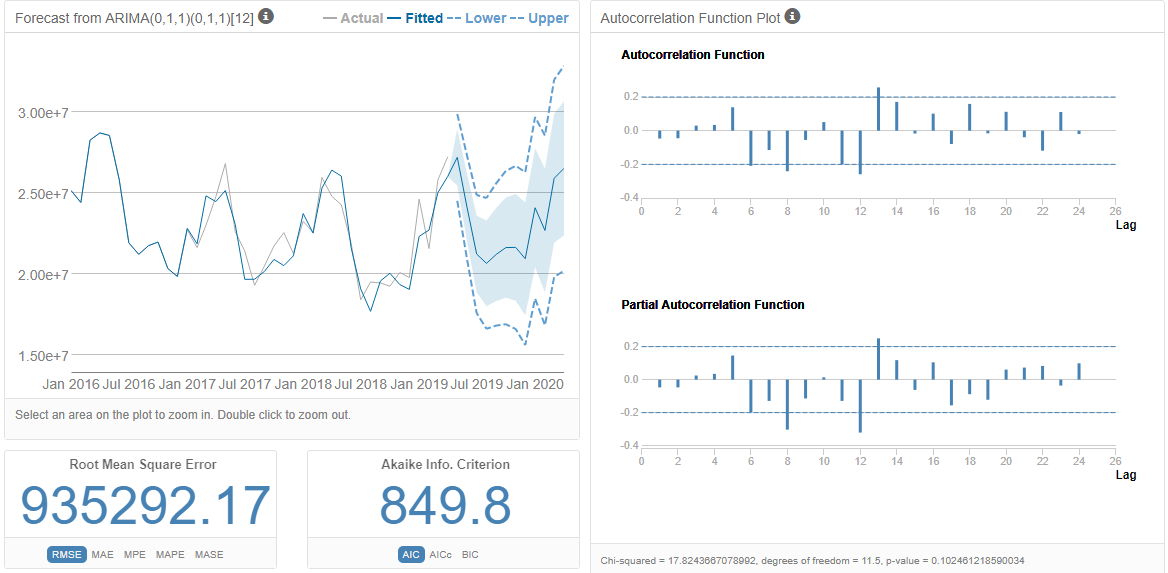
For the ARIMA model building, I brought in an ARIMA tool and connected it to the T-output of the FILTER tool just like I did for the ETS tools. In the configuration panel, I gave the model name ARIMA, set the target field as ***Sum\_Produce*** and target field frequency as monthly. Under the “model customization” tab, I did not check “Completely user specified model” (which would have meant that for the options of the non-seasonal components and the seasonal components, I would have set 0, 1, 1 in accordance with my previously determined ARIMA(0, 1, 1), but I checked “Customize the parameters used for automatic model creation”. This is so as to ensure accuracy by automating the process by letting the software choose the best ARIMA model for me, like I did when building the ETS model.

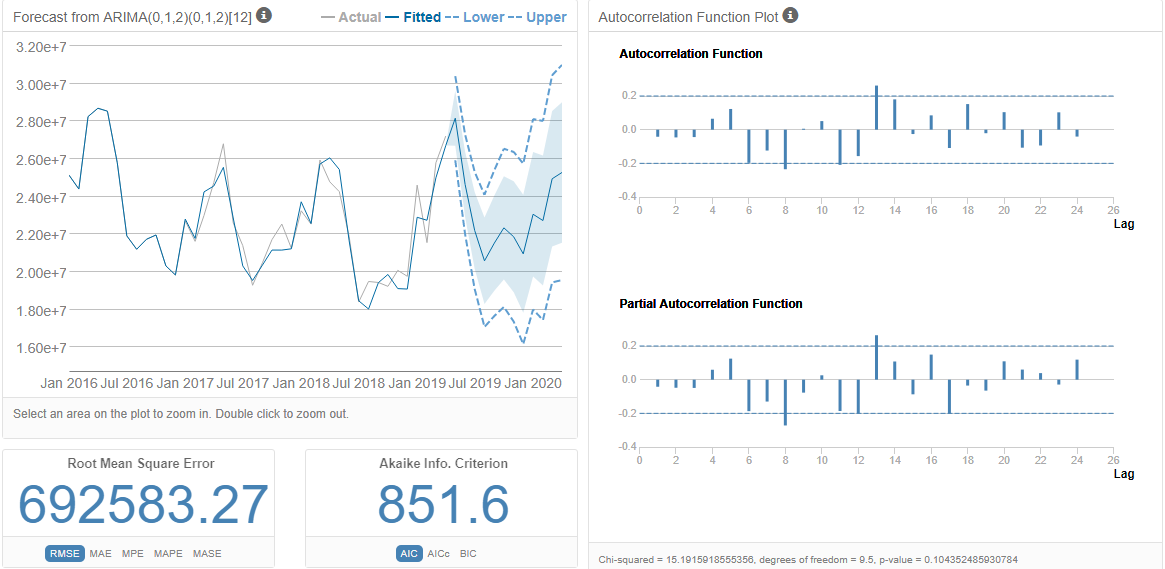
***NB: The* “Customize the parameters used for automatic model creation” *option under the “Model customization” tab can be used to automatically determine the best ARIMA format without having to use the TS PLOT tool nor going through the steps taken above. But for this project, it was required that we manually discover the best format (as it is important to get an understanding of how it works), hence the processes above.***

Now, under the “other options” tab, I set as same as I did for the ETS tool as stated above.

And now, just like I did for the ETS model, since it is wise to build more than one model type and compare the results I copied my ARIMA tool and pasted unto the canvass and named it ARIMA\_MA2, retained same configuration, except changing the “order of the seasonal moving average component” from 1 to 2, that is, ARIMA(0, 1, 2).

Finally, I added BROWSE tools to each ARIMA tool and ran the workflow to build the models.





Now, looking at the I-output results shown above of the two ARIMA tools respectively, the ARIMA with 2 MA terms is the better for the following reasons:

1. It has a much lower RMSE (Root Mean Square Error) value than the ARIMA with 1 MA term. A low RMSE value means that a forecast has a narrow range of possible values, which is a good thing.
2. It has approximately same AIC value as the other (a lower AIC actually indicates a better model)

Hence, the ARIMA\_MA2 will be used to compare with the chosen ETS configuration above in the TS COMPARE tool to choose the final best model.

**Determining the final best model – ETS or ARIMA**

After selecting one ETS and one ARIMA model, next thing I did was to determine the best forecasting model to use to forecast the next 12 months for the existing and new stores.

To make this decision, three parameters need to be considered and obtained from both the ETS and ARIMA models.

1. Residual plots
2. Forecasting errors
3. Akaike information criteria (AIC), which is a measure of the relative quality of a statistical model. A model with the lowest AIC value is considered the best fit.

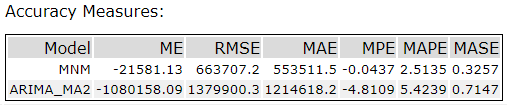
Now, one good way to validate a model’s ability to make a forecast is the use of holdout sample – a subset of the time series, usually the most recent data points, that is withheld and then used to check the accuracy of predictions from your model. The model with the lower errors has better predictive qualities and need to be used for forecasting.

Recall that when building the ETS and ARIMA models, we withheld the last 6 months/records and built the model with the remaining data.

*NB: Splitting our dataset this way is only done when building the model. When it is time to use the model to actually do the forecasts, we will use the entire dataset.*

Thus, I tested the results of both models against the data in the holdout sample using the TS COMPARE tool.

To do this, I first joined the outputs from my chosen ETS and ARIMA models using a UNION tool, and then connected its out to the L-input of a TS COMPARE tool, whilst connecting the F-output (holdout sample) of the FILTER tool to the R-input of the TS COMPARE tool. I added BROWSE tools and then ran the workflow.



Looking at the R-output of the TS COMPARE tool, we can see that the ETS(M, N, M) model has a much lower RMSE value than the ARIMA\_MA2 model, implying that it has lower errors (a variance of about 663707 units around the mean), which makes it more accurate and hence should be used for forecasting.

More so, the MASE of the ETS(M, N, M) shows a fairly strong forecast at 0.3257 with its value falling well below the generic 1.00, the commonly accepted MASE threshold for model accuracy.

Hence, I chose the ETS(M, N, M) as my overall best forecasting model.

**Forecasts for existing stores**

For existing stores, we are trying to get the monthly total for past stores before forecasting.

To begin, I opened a new canvass in alteryx and used an INPUT DATA tool to introduce the storesalesdata all over again, connected an AUTO FIELD tool, ran the workflow, after which a SELECT tool like before, and then a SUMMARIZE tool and configured exactly like before as follows:

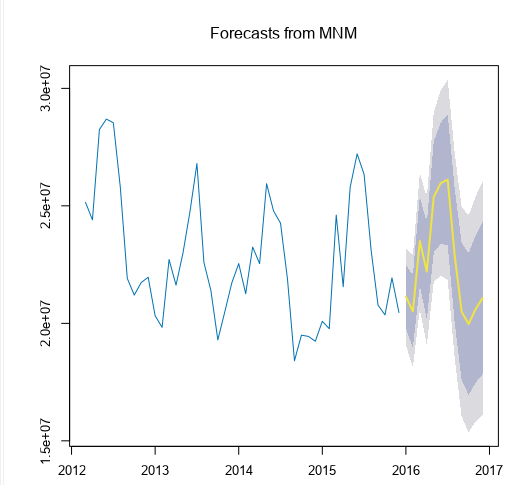
* Grouped by Year
* Grouped by Month
* Sum Produce.

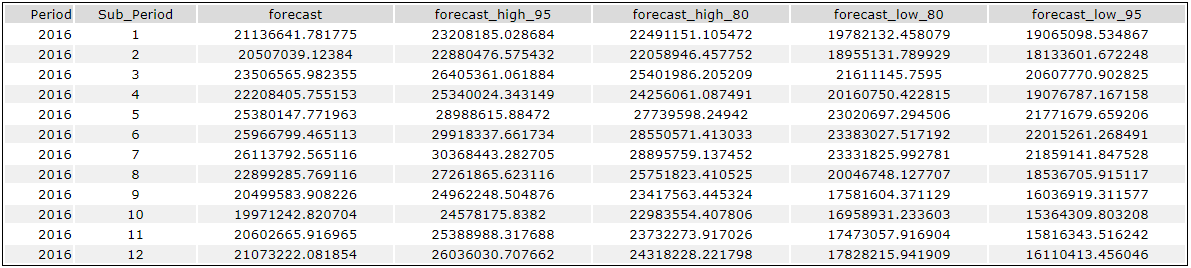
It is worthy to state at this point that the entire dataset is being used to forecast, unlike before.

Next, I copied and pasted the ETS(M, N, M) from the previous workflow. However, in the configuration panel I set the series starting date to be year 2012 and month 3 since that is the starting date of the storesalesdata file.

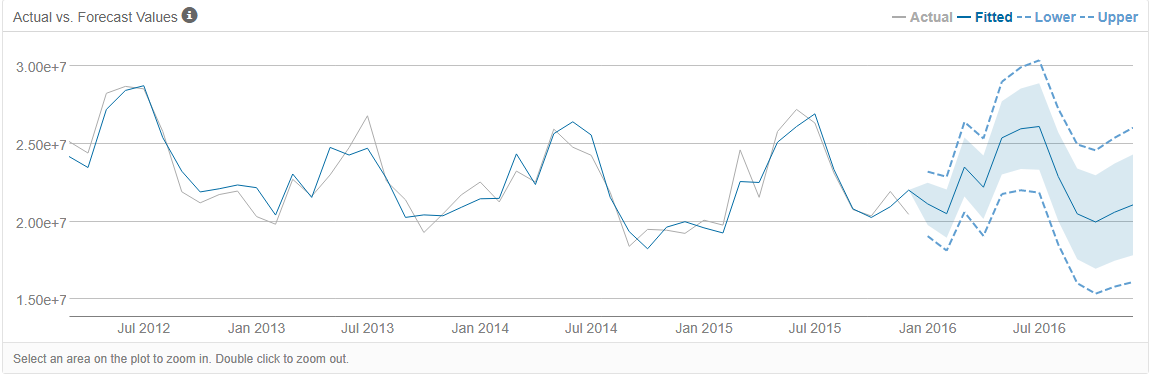
Then I connected a TS FORECAST tool (which can provide forecast for either an ETS or ARIMA model for a specified number of periods) to its output, and in the configuration panel I left the default generated “field name for the point forecast” as ***forecast*** and set the “number of periods into the future to forecast” as 12***.*** Then I connected an OUTPUT DATA tool to the O-output of the TS FORECAST tool in order to save the file whilst connecting BROWSE tools to the other outputs, and then ran the workflow.

The results from the R- and I-outputs are shown below

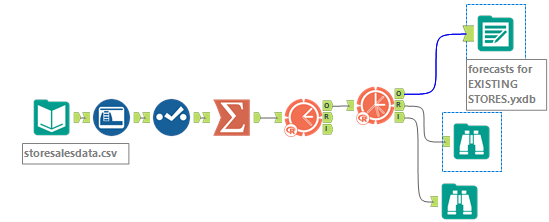




And shown graphically below.



The associated workflow is shown below.



**Forecasts for new stores**

For new stores we are trying to get the average monthly total of a store per cluster.

Now, since there is no previous sales data for the new stores, we will forecast the produce sales for the clusters each new store belongs to and assume these to be the predicted forecasts for each new store.

We will be forecasting for each of the clusters (as can be found in the preparedplusclusterID dataset from task 1 above) and then multiplying the results by the number of new stores in that cluster. Then we will sum the new stores produce sales forecasts for each of the segments to get the forecast for all new stores, that is, we will be adding all of these forecasts together on the same months to get a total forecast for all the new stores. This we will accomplish by adding a UNION tool and a SUMMARIZE tool.

Thus, I began by introducing storesales dataset and my earlier generated preparedplusclusterID dataset for new stores into a new canvass in alteryx using two INPUT DATA tools. Next, I connected two AUTO FIELD tools, one for each dataset, and ran the workflow to ensure each field is in its correct data type. As usual, I used a SELECT tool to ensure the ***year, month*** and ***day*** fields are in string format. Then I used a JOIN tool to combine both datasets, joining on ***store*** field.

Then, I connected a SUMMARIZE tool to the J-output and

* Grouped by Store
* Grouped by Cluster
* Grouped by Year
* Grouped by Month
* Sum Produce.

And then, connecting another SUMMARIZE tool, I

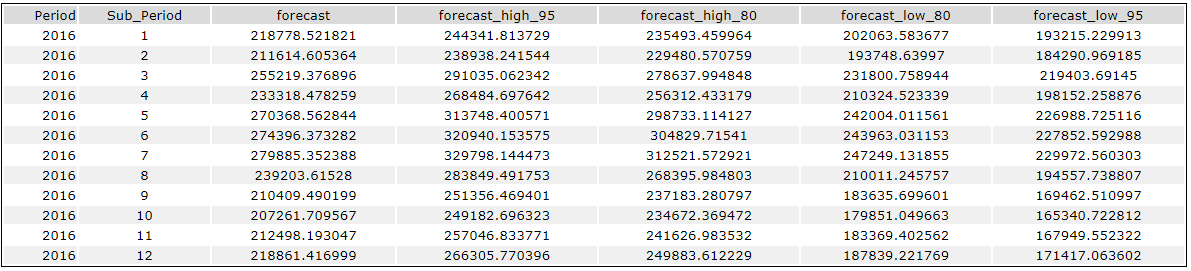
* Grouped by Cluster
* Grouped by Year
* Grouped by Month
* Avg Sum\_Produce.

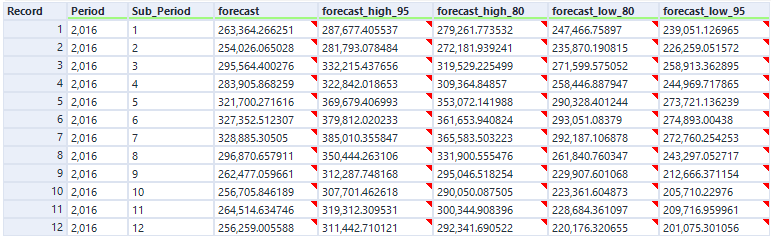
It is worthy to state at this point that the entire dataset is being used to forecast, just like we did for existing stores.

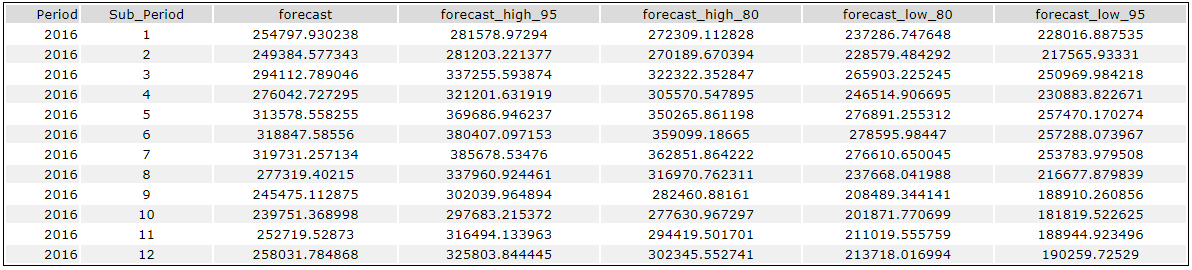
Next, I introduced three FILTER tools to filter each of the three clusters/segments, so that we can determine the forecast for each specific cluster. Then I copied and pasted three identical ETS models with same configuration as used for the existing stores forecast (except setting the “target field” as Avg\_Sum\_Produce) to each of the T-outputs of the FILTER tools in order to get the actual forecasts per cluster.

Then I connected TS FORECAST tools (which can provide forecast for either an ETS or ARIMA model for a specified number of periods) to each output, and in the configuration panel I left the default generated “field name for the point forecast” as ***forecast*** and set the “number of periods into the future to forecast” as 12***.***

The forecasts for cluster 1, 2 and 3 are shown respectively below.







As seen above, we now have the forecasts for each of the 3 clusters.

Next, I used three FORMULA tools each to multiply the above results by the number of new stores in each cluster so we can obtain the total forecast for each segment for the new stores. As can be seen from the PREDICTED CLUSTERS for new stores dataset already prepared above in task 2, cluster 1 has only one store, cluster 2 six stores while cluster 3 has three stores (making a total of 10 new stores).

Thus, in the FORMULA tool, I multiplied the forecast for cluster 1 by 1, that for cluster 2 by 6 and that for cluster 3 by 3.

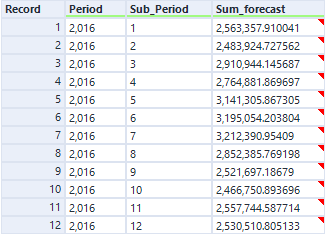
Next, I connected the outputs of all three FORMULA tools to a UNION tool so that they can together be fed into a SUMMARIZE tool so as to obtain the sum of the new stores produce sales forecasts for each of the clusters which gives us the forecasts for all the new stores.

In the SUMMARIZE tool, I

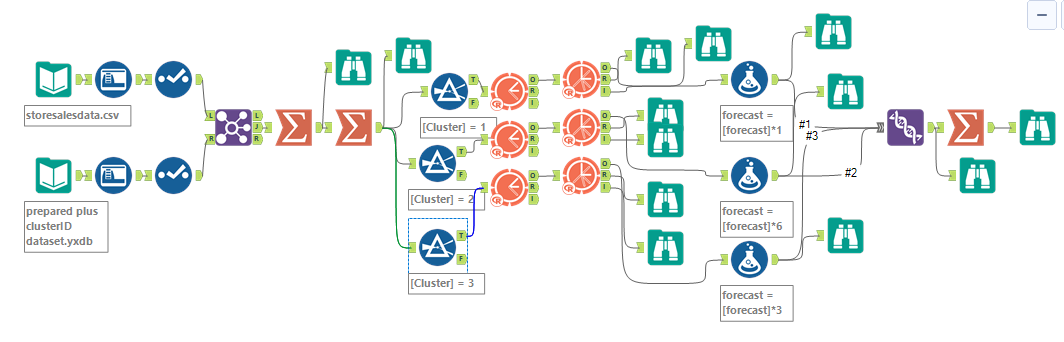
* Grouped by ***period***
* Grouped by ***Sub-period***
* Sum ***forecast***

Then I added a BROWSE tool and ran the workflow.

The table showing the total forecasts for the new stores for the entire period of 2016 is shown below.



The associated workflow is shown below



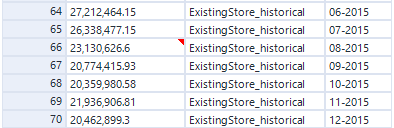
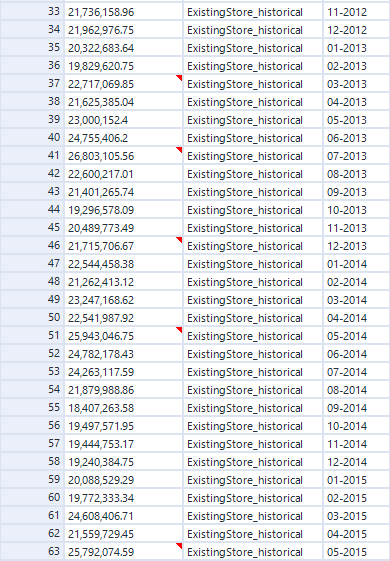
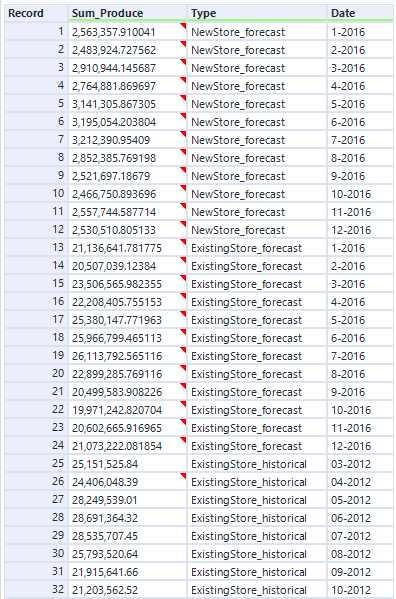
**Visualization of forecasts**

In order to stack up my dataset to produce visualization in Tableau, I used three INPUT tools to introduce my already prepared **forecasts for new stores, forecasts for existing stores** and the given **storesales** datasets onto a new canvass in alteryx.

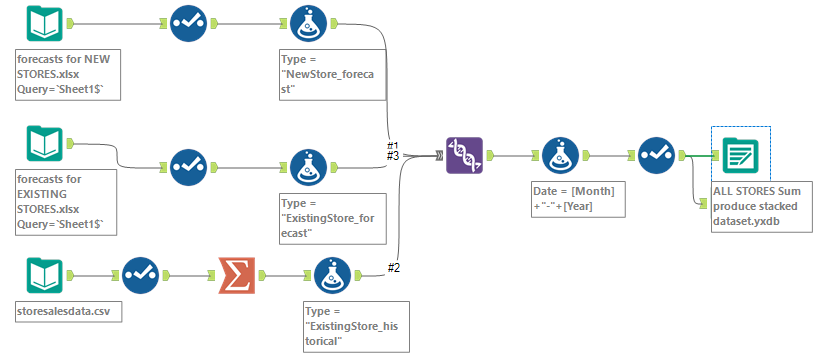
Next, I used SELECT tools to change the names of such fields like ***period, Sub-period*** and ***forecast*** to ***Year, Month*** and ***Sum\_Produce*** respectively. Meanwhile, for the **storesales** dataset, I used a SUMMARISE tool to Group by Year, Month and then I Summed Produce. I then used three formula tools to create a new field/column called ***Type*** (with which I can differentiate new stores forecast, existing stores forecasts and existing stores historical sales information).

Then, I used a UNION tool to stack up the information in the three datasets whilst deselecting irrelevant fields. Next I used a FORMULA tool to create a new field ***Date*** to merge the ***Year*** and ***Month*** columns into a single date field, via using the formula: [Month]+"-"+[Year]. Finally, I used a SELECT tool to deselect the irrelevant columns, connected an OUTPUT DATA tool to save the dataset.

The resulting output of the BROWSE tool added is shown below.



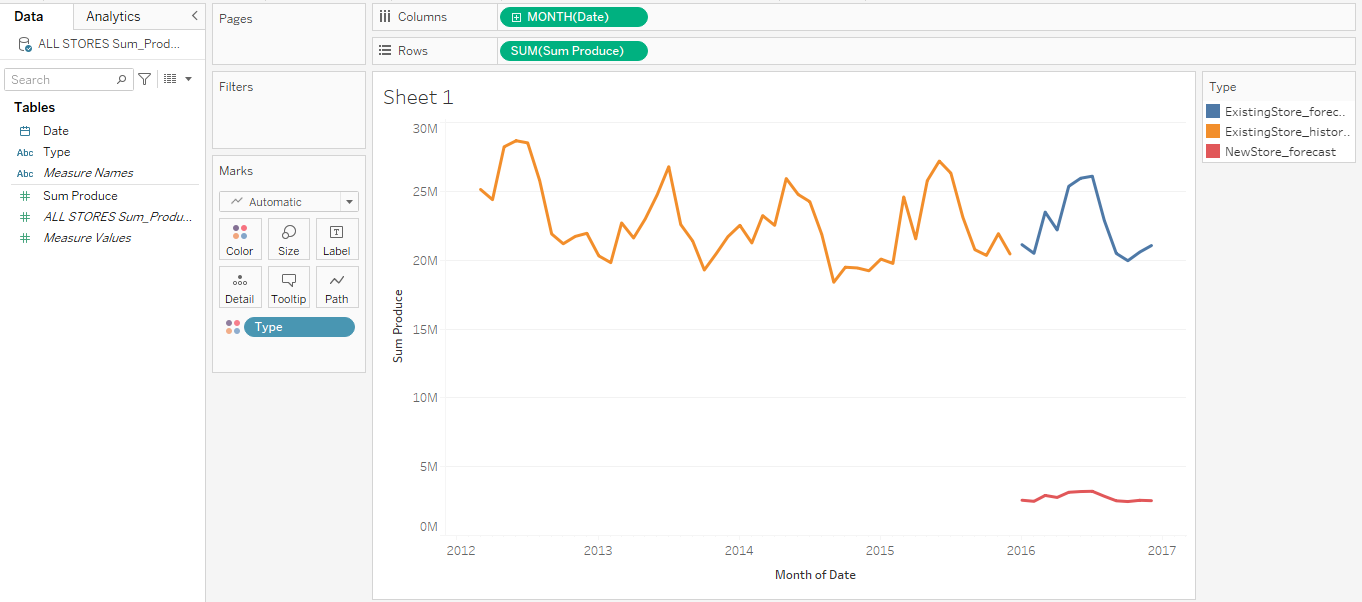
The associated workflow is shown below



The above dataset was then imported to Tableau.

Once brought into Tableau, I used ***Date*** and ***Sum\_Produce*** in my “Columns” and “Rows” respectively. I next changed the field type of ***Date*** to Date type and then selected Month. Finally, I coloured by type.

My result is shown below.



Before you submit

Please check your answers against the requirements of the project dictated by the rubric. Reviewers will use this rubric to grade your project.