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

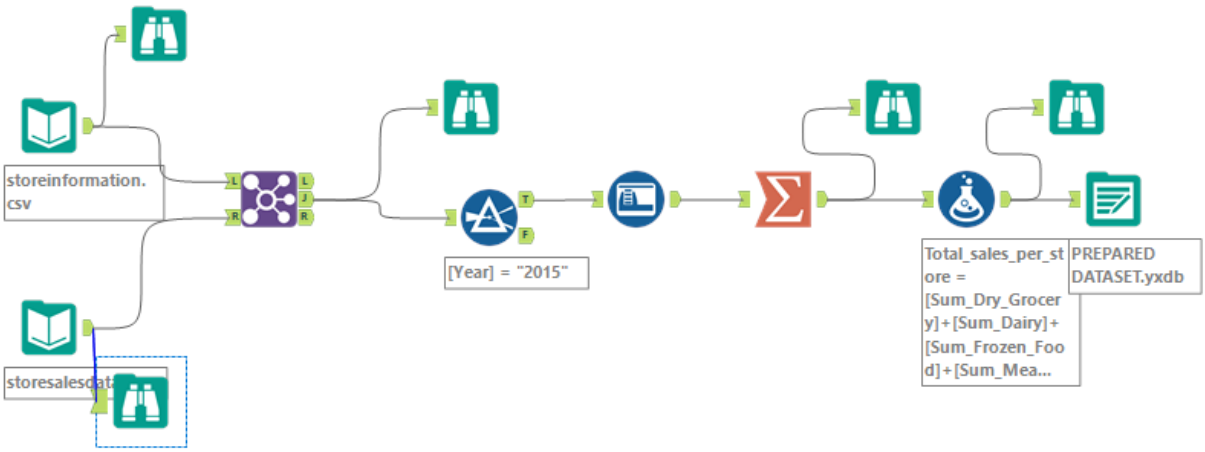
The optimal number of store formats is 3.

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.



2. 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.

K-Means Cluster Assessment Report

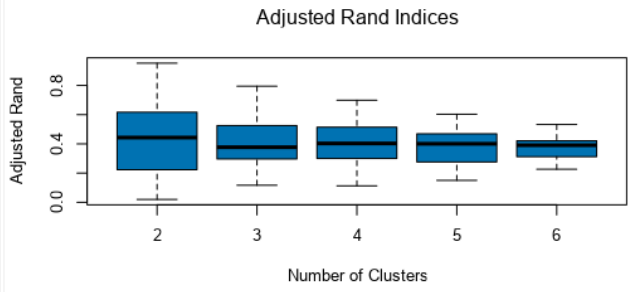
Summary Statistics

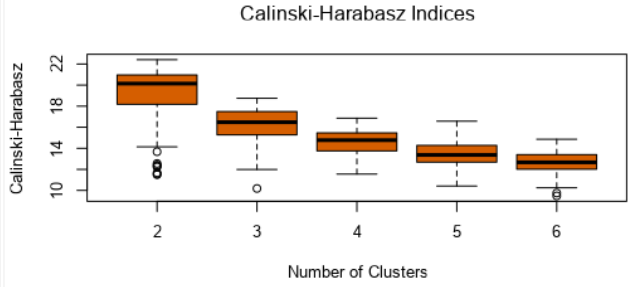
Adjusted Rand Indices:

	2	3	4	5	6
Minimum	0.020011	0.116722	0.113112	0.150418	0.226486
1st Quartile	0.225885	0.297259	0.300394	0.278121	0.31428
Median	0.443086	0.377155	0.403137	0.400518	0.390769
Mean	0.430858	0.421041	0.403641	0.3825	0.377712
3rd Quartile	0.607523	0.525493	0.511782	0.468717	0.421245
Maximum	0.952115	0.794667	0.698785	0.602951	0.532821

Calinski-Harabasz Indices:

	2	3	4	5	6
Minimum	11.49694	10.18405	11.55369	10.41516	9.47976
1st Quartile	18.25951	15.28341	13.77306	12.69596	12.01426
Median	20.14522	16.47868	14.77552	13.36888	12.66809
Mean	19.09552	16.27797	14.57626	13.43979	12.64158
3rd Quartile	20.94642	17.45689	15.46508	14.25764	13.39232
Maximum	22.41555	18.75042	16.86351	16.57168	14.8625





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

K-Medians Cluster Assessment Report

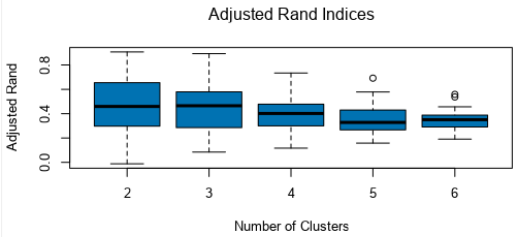
Summary Statistics

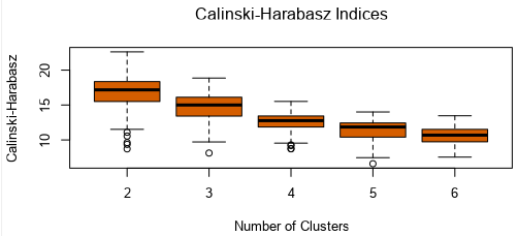
Adjusted Rand Indices:

	2	3	4	5	6
Minimum	-0.011625	0.084526	0.116506	0.158136	0.190645
1st Quartile	0.297396	0.286042	0.301225	0.267286	0.291632
Median	0.459513	0.465229	0.401731	0.328073	0.350731
Mean	0.466856	0.439761	0.404686	0.35267	0.344482
3rd Quartile	0.654912	0.578325	0.476819	0.428266	0.387178
Maximum	0.907003	0.892867	0.73414	0.691883	0.558439

Calinski-Harabasz Indices:

	2	3	4	5	6
Minimum	8.719918	8.123626	8.731078	6.595262	7.541222
1st Quartile	15.54369	13.427906	11.864249	10.403504	9.739462
Median	17.175761	14.998873	12.758965	11.847889	10.685637
Mean	16.675595	14.621134	12.515871	11.4006	10.637912
3rd Quartile	18.350376	16.106787	13.413412	12.417842	11.500562
Maximum	22.607996	18.857375	15.514958	14.009551	13.460598





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.

Neural Gas Cluster Assessment Report

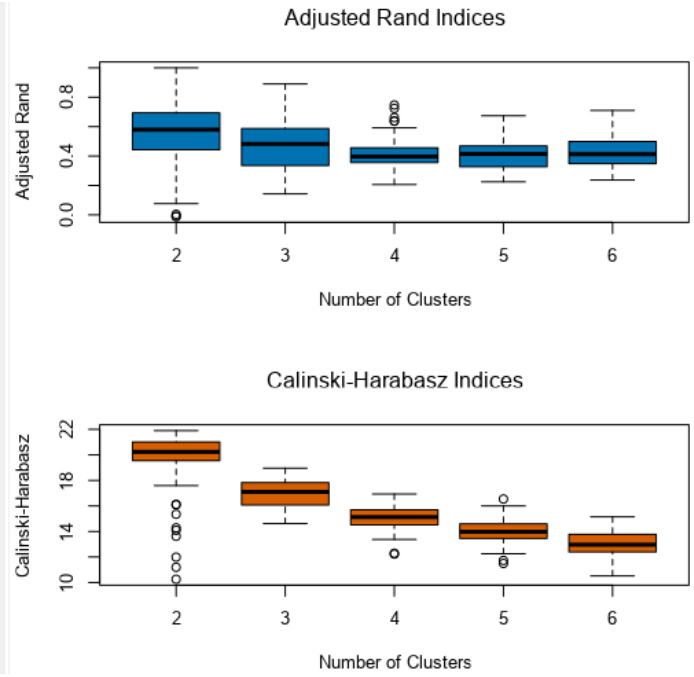
Summary Statistics

Adjusted Rand Indices:

	2	3	4	5	6
Minimum	-0.010865	0.142572	0.206649	0.224906	0.236978
1st Quartile	0.451217	0.33785	0.356678	0.330361	0.350558
Median	0.579839	0.482362	0.397026	0.414396	0.412798
Mean	0.557849	0.471379	0.416606	0.409261	0.432222
3rd Quartile	0.694198	0.57865	0.455998	0.469004	0.496961
Maximum	1	0.890791	0.74727	0.67417	0.710526

Calinski-Harabasz Indices:

	2	3	4	5	6
Minimum	10.25416	14.61915	12.22926	11.49669	10.51791
1st Quartile	19.55429	16.08053	14.52528	13.46469	12.4012
Median	20.23337	17.09897	15.12798	13.97049	12.96769
Mean	19.71051	16.90691	15.04074	14.01655	13.08207
3rd Quartile	21.01077	17.82778	15.69566	14.60431	13.7743
Maximum	21.89885	18.96107	16.93614	16.54468	15.14481



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.

Summary Report of the K-Means Clustering Solution ClusterBySales_Data

Solution Summary

Call:

stepFlexclust(scale(model.matrix(~-1 + Percent_sales_Dry_Grocery + Percent_sales_Diary + Percent_sales_Frozen_Food + Percent_sales_Meat + Percent_sales_Produce + Percent_sales_Floral + Percent_sales_Deli + Percent_sales_Bakery + Percent_sales_General_Merchandise, the.data)), k = 3, nrep = 6, FUN = kcca, family = kccaFamily("kmeans"))

Cluster Information:

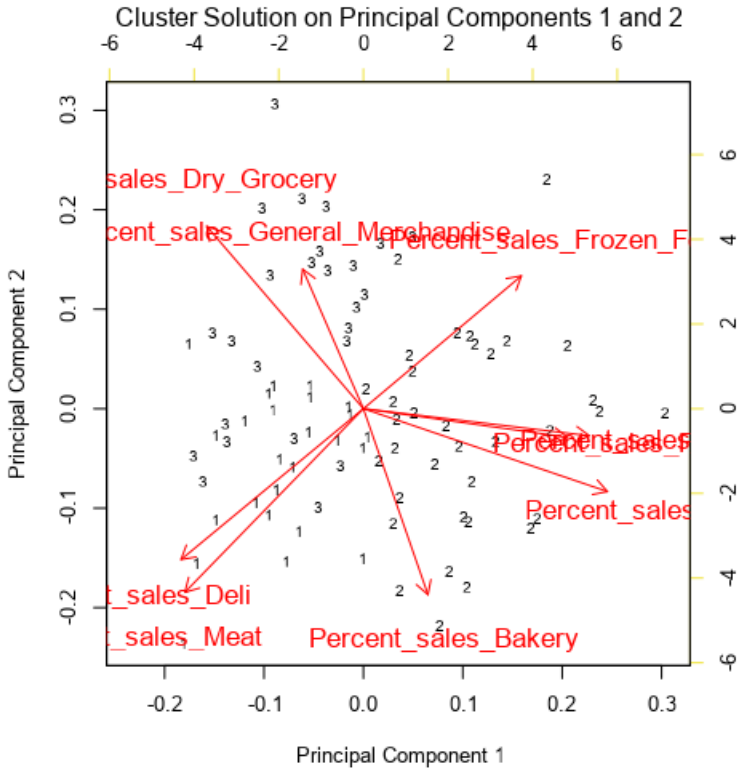
Cluster	Size	Ave Distance	Max Distance	Separation
1	25	2.099985	4.823871	2.191566
2	35	2.475018	4.412367	1.947298
3	25	2.289004	3.585931	1.72574

Convergence after 8 iterations.

Sum of within cluster distances: 196.35034.

	Percent_sales_Dry_Grocery	Percent_sales_Diary	Percent_sales_Frozen_Food	Percent_sales_Meat	Percent_sales_Produce	Percent_sales_Floral	Percent_sales_Deli
1	0.528249	-0.215879	-0.261597	0.614147	-0.655027	-0.663872	0.824834
2	-0.594802	0.655893	0.435129	-0.384631	0.812883	0.71741	-0.46168
3	0.304474	-0.702372	-0.347583	-0.075664	-0.483009	-0.340502	-0.178481

	Percent_sales_Bakery	Percent_sales_General_Merchandise
1	0.428226	-0.674769
2	0.312878	-0.329045
3	-0.866255	1.135432



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.

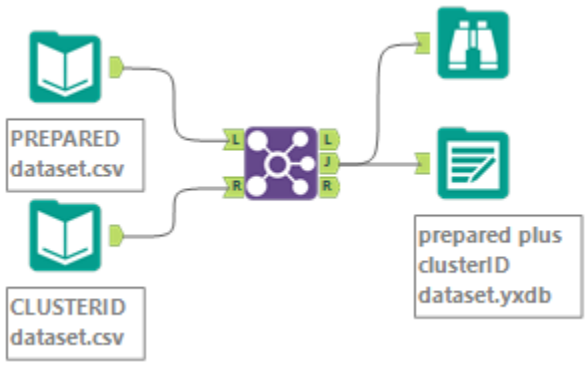
The 2nd chart in the summary report shows the inter-cluster comparisons for each variable. For instance, for ***Percent_sales_Dry_Grocery***, cluster 1 has the highest positive value while cluster 2 has the highest negative value. This implies that the stores in cluster 1 are the greatest opposites to 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**

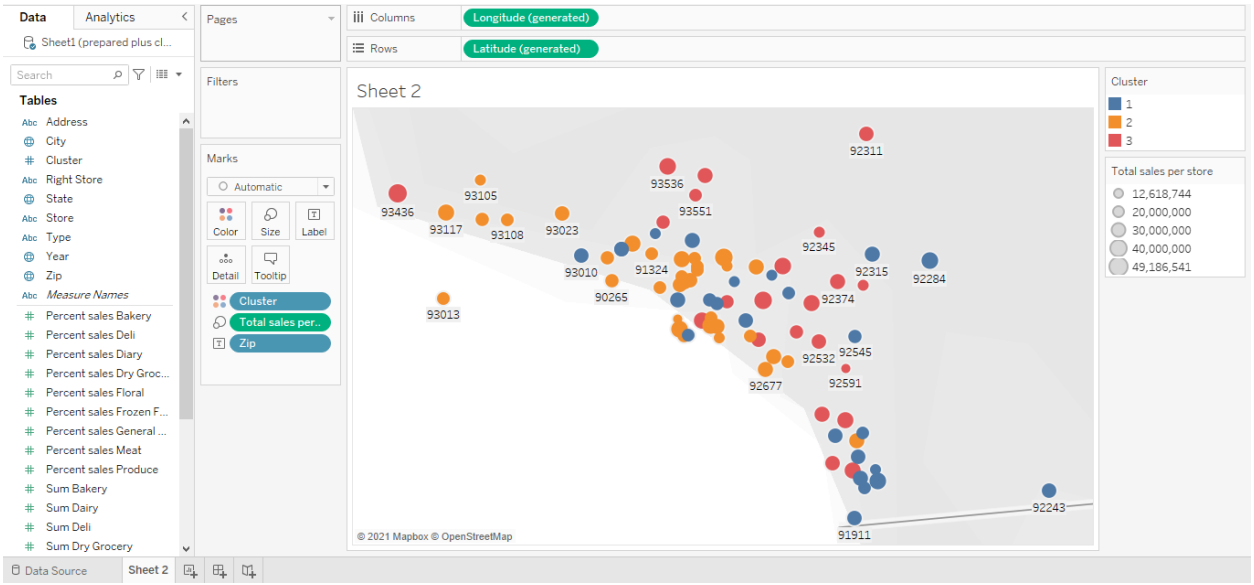
To provide some visualisations in Tableau, I used a JOIN tool in alteryx to combine both my raw PREPARED dataset and my CLUSTERID dataset and then saved as a csv so that it can be accessed by Tableau to generate some visualizations to see if my solution makes sense.



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.

Record	Model	Accuracy	Accuracy_1	Accuracy_2	Accuracy_3	F1
1	DecisionTree_CAPSTONE	0.647059	0.5	1	0.5	0.666667
2	BoostedModel_CAPSTONE	0.764706	0.5	1	1	0.833333
3	ForestModel_CAPSTONE	0.705882	0.5	1	0.75	0.75

Record	Actual	DecisionTree_CAPSTONE	BoostedModel_CAPSTONE	ForestModel_CAPSTONE
1	1	1	1	1
2	1	2	1	1
3	1	2	2	2
4	2	2	2	2
5	1	1	3	2
6	2	2	2	2
7	1	1	1	3
8	3	1	3	1
9	2	2	2	2
10	1	1	1	1
11	3	3	3	3
12	1	2	2	1
13	1	3	3	3
14	2	2	2	2
15	3	1	3	3
16	2	2	2	2
17	3	3	3	3

Model Comparison Report

Fit and error measures

Model	Accuracy	F1	Accuracy_1	Accuracy_2	Accuracy_3
DecisionTree_CAPSTONE	0.6471	0.6667	0.5000	1.0000	0.5000
BoostedModel_CAPSTONE	0.7647	0.8333	0.5000	1.0000	1.0000
ForestModel_CAPSTONE	0.7059	0.7500	0.5000	1.0000	0.7500

Model: model names in the current comparison.
Accuracy: overall accuracy, number of correct predictions of all classes divided by total sample number.
Accuracy_[class name]: accuracy of Class [class name] is defined as the number of cases that are **correctly** predicted to be Class [class name] divided by the total number of cases that actually belong to Class [class name], this measure is also known as *recall*.
AUC: area under the ROC curve, only available for two-class classification.
F1: F1 score, 2 * precision * recall / (precision + recall). The *precision* measure is the percentage of actual members of a class that were predicted to be in that class divided by the total number of cases predicted to be in that class. In situations where there are three or more classes, average precision and average recall values across classes are used to calculate the F1 score.

Confusion matrix of BoostedModel_CAPSTONE

	Actual_1	Actual_2	Actual_3
Predicted_1	4	0	0
Predicted_2	2	5	0
Predicted_3	2	0	4

Confusion matrix of DecisionTree_CAPSTONE

	Actual_1	Actual_2	Actual_3
Predicted_1	4	0	2
Predicted_2	3	5	0
Predicted_3	1	0	2

Confusion matrix of ForestModel_CAPSTONE

	Actual_1	Actual_2	Actual_3
Predicted_1	4	0	1
Predicted_2	2	5	0
Predicted_3	2	0	3

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

Record	er	PopPacIsl	PopWhite	HVal0to100K	HVal100Kto200K	HVal200Kto300K	HVal300Kto400K	HVal400Kto500K	HVal500Kto750K	HVal750KPlus	PopDens	Score_1	Score_2	Score_3
1	6	0.000756	0.179619	0.130383	0.13756	0.088517	0.113038	0.121411	0.325359	0.083732	2,094.407018	0.614113	0.049708	0.33618
2	4	0.003119	0.506368	0.017774	0.018442	0.092075	0.113324	0.09702	0.34545	0.315916	6,256.72792	0.1022	0.795133	0.102667
3	3	0.004828	0.043139	0.047129	0.028211	0.094922	0.155659	0.13309	0.375705	0.165284	8,043.562891	0.245342	0.227712	0.526945
4		0.005308	0.453006	0.035694	0.060048	0.060978	0.080312	0.068786	0.415691	0.27849	7,547.025711	0.015678	0.966272	0.01805
5	3	0.00659	0.527099	0.022281	0.019634	0.017648	0.070152	0.054269	0.291198	0.524818	7,621.043926	0.021153	0.962402	0.016445
6	8	0.006286	0.405789	0.101091	0.210909	0.368727	0.167273	0.044364	0.074909	0.032727	1,054.522398	0.033716	0.002932	0.963353
7	5	0.001625	0.471739	0.027035	0.048877	0.137926	0.136857	0.151062	0.35711	0.141133	8,639.436528	0.013225	0.966806	0.019969
8		0.004384	0.469771	0.013704	0.192849	0.346456	0.197459	0.108135	0.112744	0.028653	3,207.438094	0.022989	0.006037	0.970974
9	8	0.001893	0.713645	0.008479	0.019272	0.121025	0.162074	0.099827	0.189632	0.399692	4,435.823519	0.007082	0.987997	0.004921
10	5	0.002157	0.567129	0.196016	0.053418	0.183794	0.184699	0.189679	0.181077	0.011317	2,663.834099	0.090802	0.857985	0.051213

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.

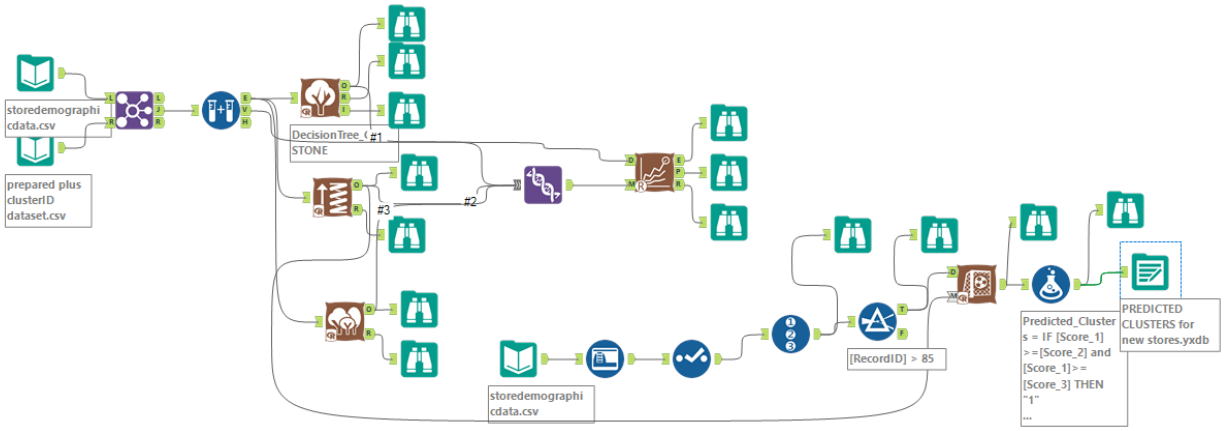
Record	opWhite	HVal0to100K	HVal100Kto200K	HVal200Kto300K	HVal300Kto400K	HVal400Kto500K	HVal500Kto750K	HVal750KPlus	PopDens	Score_1	Score_2	Score_3	Predicted_Clusters
1	179619	0.130383	0.13756	0.088517	0.113038	0.121411	0.325359	0.083732	2,094.407018	0.614113	0.049708	0.33618	1
2	506368	0.017774	0.018442	0.092075	0.113324	0.09702	0.34545	0.315916	6,256.72792	0.1022	0.795133	0.102667	2
3	043139	0.047129	0.028211	0.094922	0.155659	0.13309	0.375705	0.165284	8,043.562891	0.245342	0.227712	0.526945	3
4	453006	0.035694	0.060048	0.060978	0.080312	0.068786	0.415691	0.27849	7,547.025711	0.015678	0.966272	0.01805	2
5	527099	0.022281	0.019634	0.017648	0.070152	0.054269	0.291198	0.524818	7,621.043926	0.021153	0.962402	0.016445	2
6	405789	0.101091	0.210909	0.368727	0.167273	0.044364	0.074909	0.032727	1,054.522398	0.033716	0.002932	0.963353	3
7	471739	0.027035	0.048877	0.137926	0.136857	0.151062	0.35711	0.141133	8,639.436528	0.013225	0.966806	0.019969	2
8	469771	0.013704	0.192849	0.346456	0.197459	0.108135	0.112744	0.028653	3,207.438094	0.022989	0.006037	0.970974	3
9	713645	0.008479	0.019272	0.121025	0.162074	0.099827	0.189632	0.399692	4,435.823519	0.007082	0.987997	0.004921	2
10	567129	0.196016	0.053418	0.183794	0.184699	0.189679	0.181077	0.011317	2,663.834099	0.090802	0.857985	0.051213	2

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.

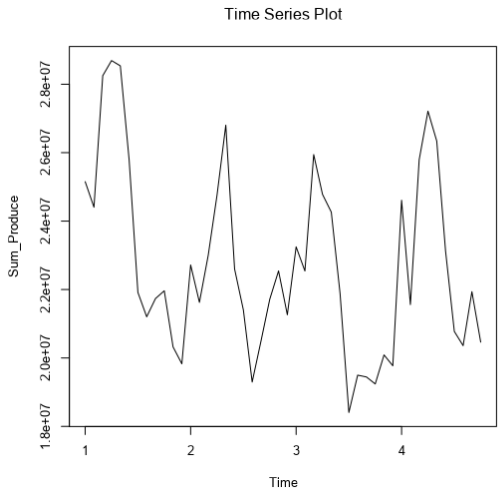
Determining the model type

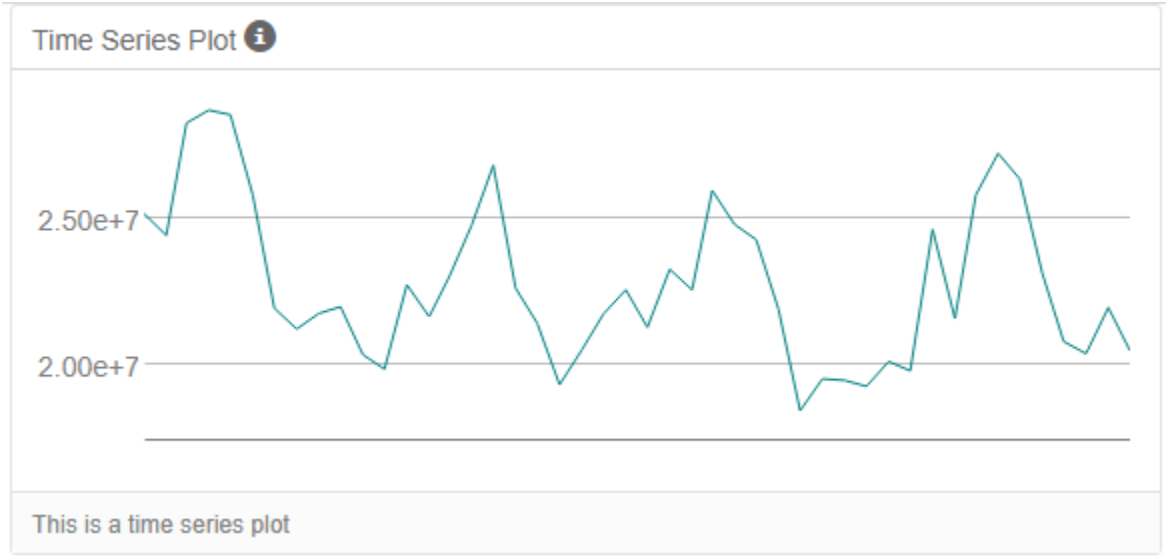
I used the INPUT DATA tool to introduce the storesalesdata into a new canvass in alteryx. Then I connected an AUTO FIELD tool to convert to appropriate data formats (since it's a csv file), ran the workflow to effect the conversions and then connected a SELECT tool to confirm the changes.

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

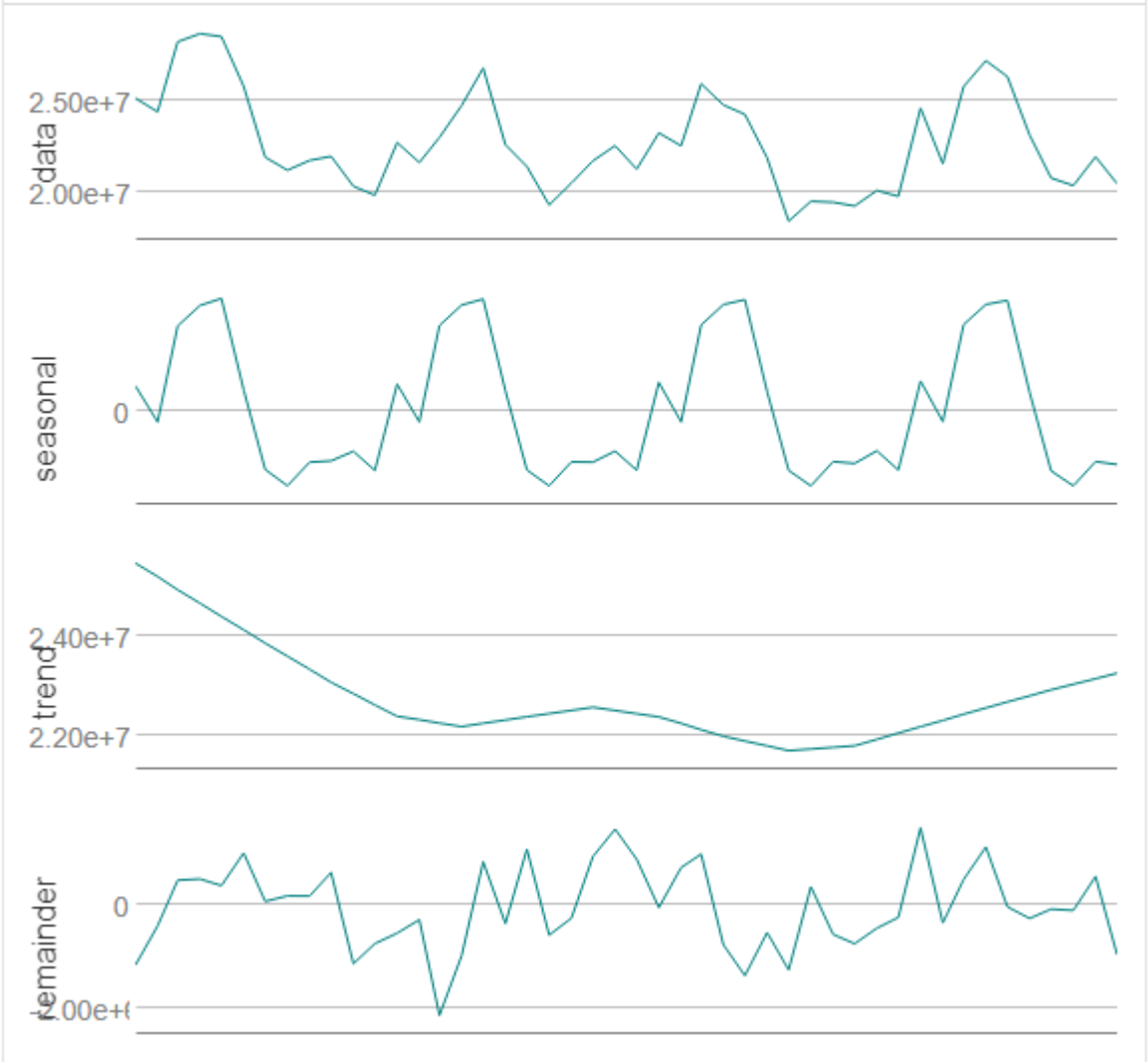
- Grouped by Year
- Grouped by Month, and then
- Sum Produce.

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.

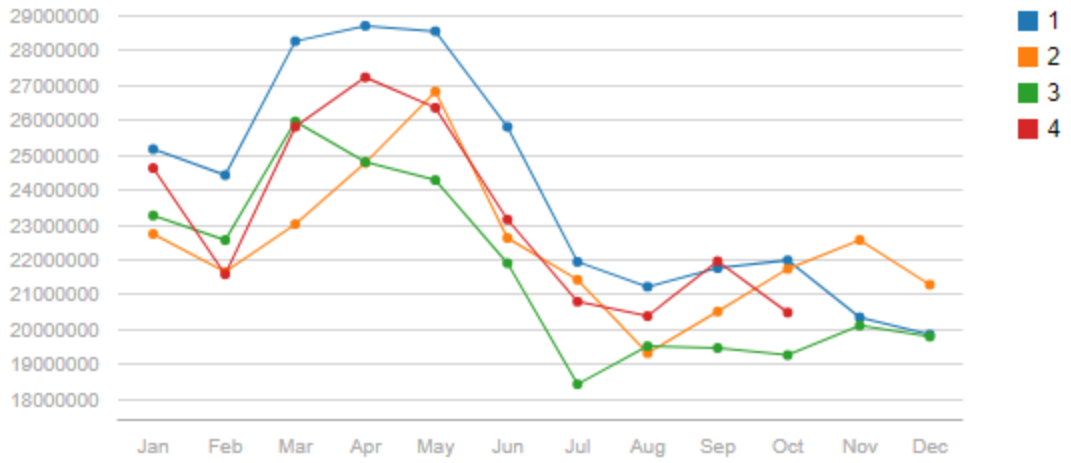




Decomposition Plot

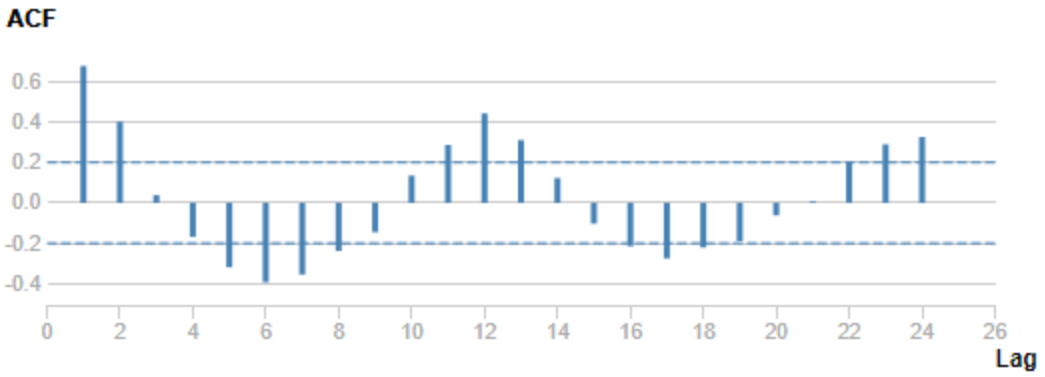


Seasonplot

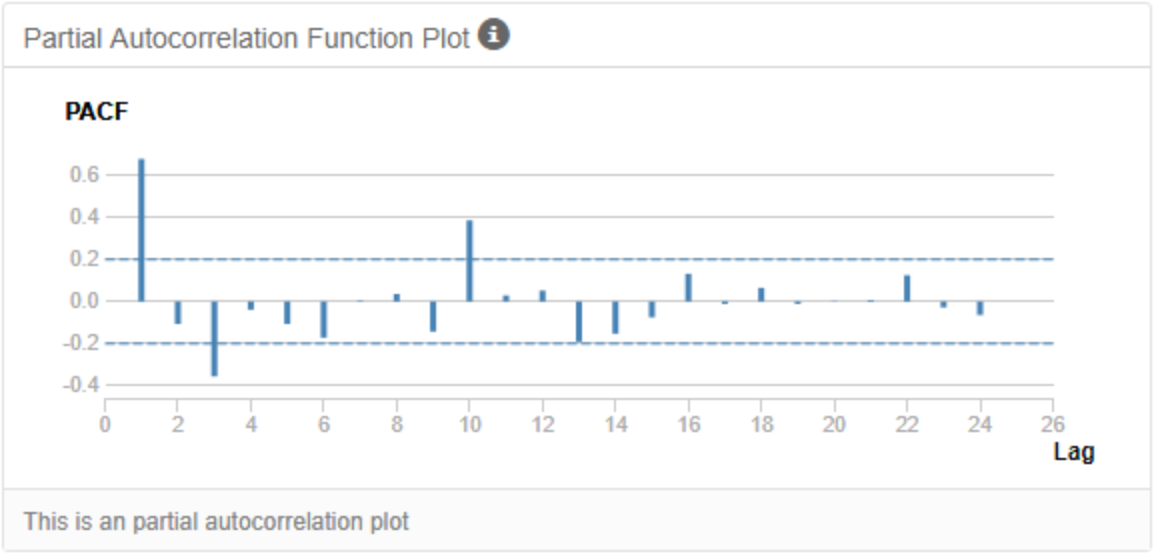


This is a season plot

Autocorrelation Function Plot



This is an autocorrelation plot



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.

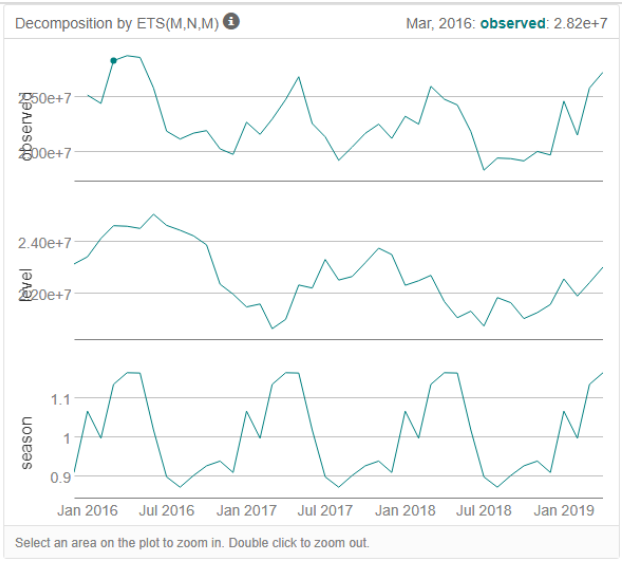
NB: The "Auto" options under the model type tab can be used to automatically determine the best ETS format without having to use the TS PLOT tool nor going through the steps taken above. But for this project, it was required that we manually discover the best format (as it is important to get an understanding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

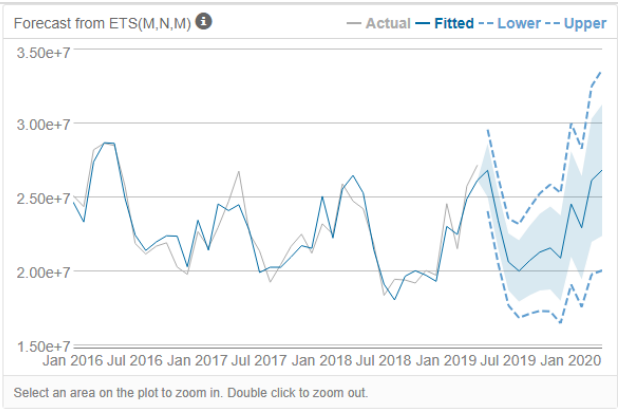
Now, under the "other options" tab, I chose 12 as the number of forecast periods (since we were asked to prepare a monthly forecast for produce sales for the full year of 2016, that is 12 months, for both existing and new stores.), and then set the "series starting period" to 2016 period 1 (that is,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

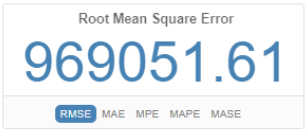
Now, to test if Trend dampening would have a positive effect on our model, I copied and pasted the ETS tool unto the canvass, connected it to the T-ouput of the FILTER tool like before and ensured same configuration settings, except checking Yes for the "Trend dampening" option plus adding the word Dampened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

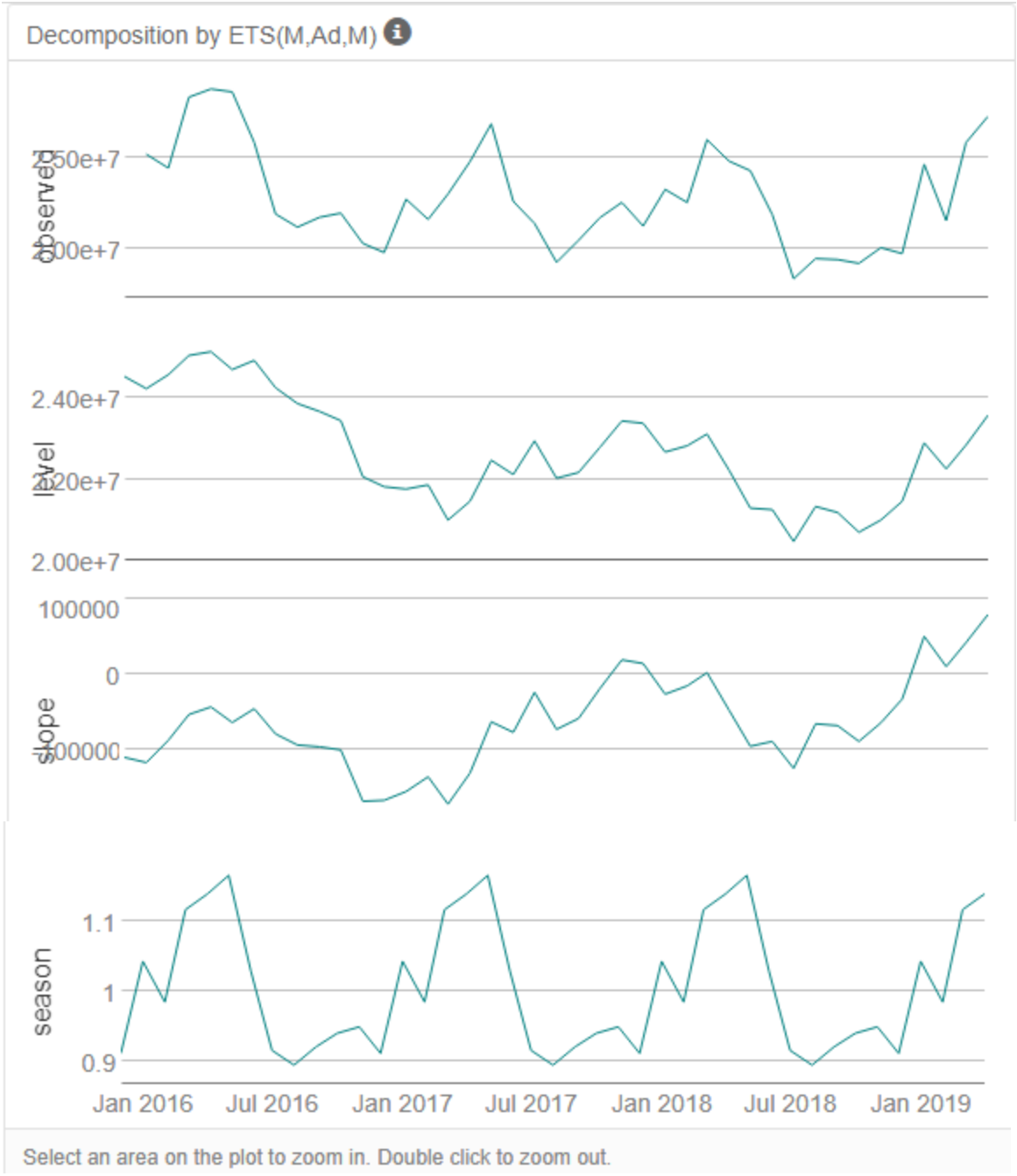
The R-output of the ETS models showed me some important information, such as in-sample air measures, information criteria and the selected alpha value.


Decomposition by ETS(M,N,M)
Mar, 2016: observed: 2.82e+7
observed
3.50e+7
3.00e+7
level
2.40e+7
2.20e+7
season
1.1
1
0.9
Jan 2016
Jul 2016
Jan 2017
Jul 2017
Jan 2018
Jul 2018
Jan 2019
Select an area on the plot to zoom in. Double click to zoom out.

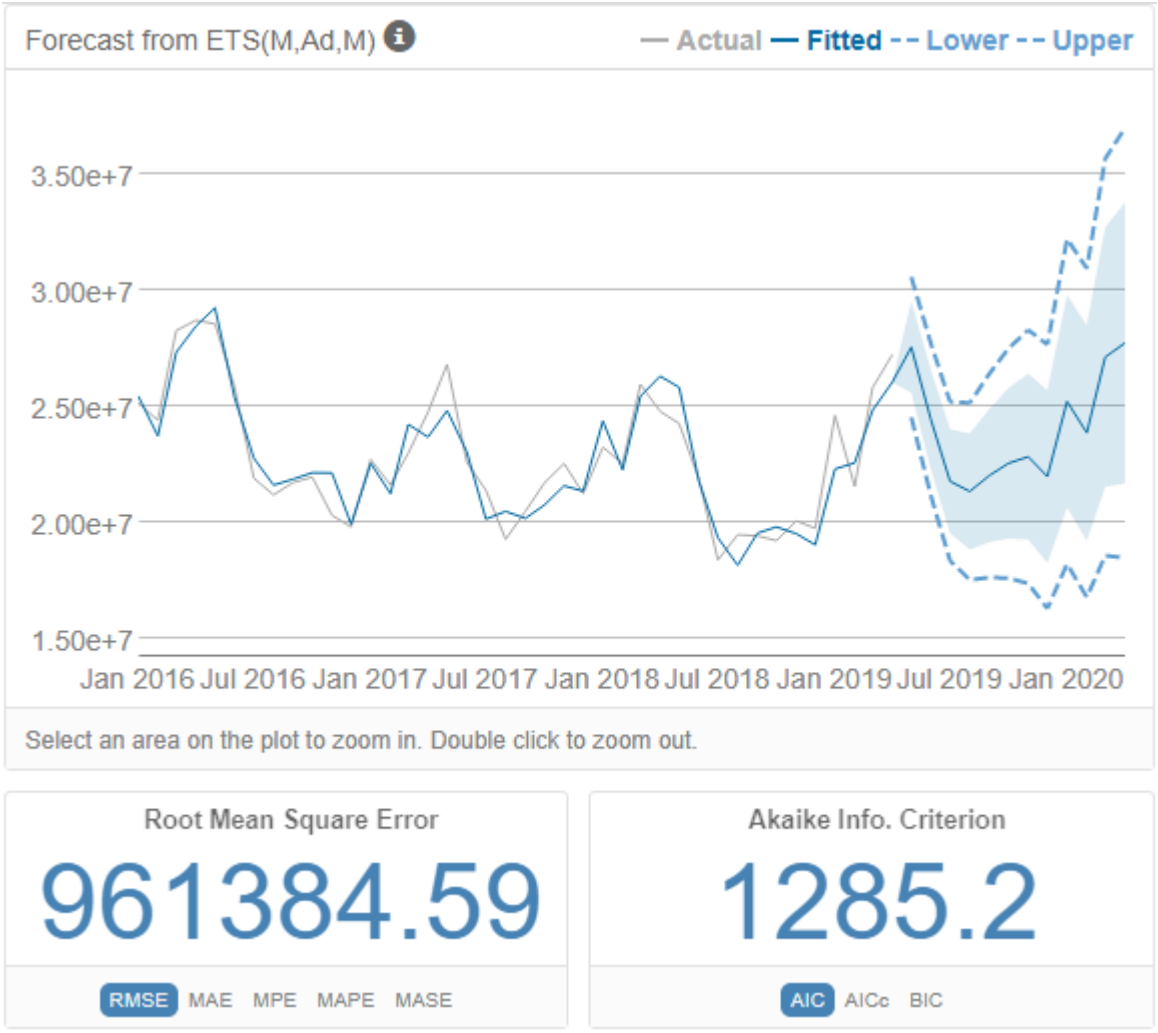

Forecast from ETS(M,N,M)
— Actual — Fitted -- Lower -- Upper
3.50e+7
3.00e+7
2.50e+7
2.00e+7
1.50e+7
Jan 2016
Jul 2016
Jan 2017
Jul 2017
Jan 2018
Jul 2018
Jan 2019
Jul 2019
Jan 2020
Select an area on the plot to zoom in. Double click to zoom out.


Root Mean Square Error
969051.61
RMSE MAE MPE MAPE MASE


Akaike Info. Criterion
1279.4
AIC AICc BIC



Select an area on the plot to zoom in. Double click to zoom out.



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

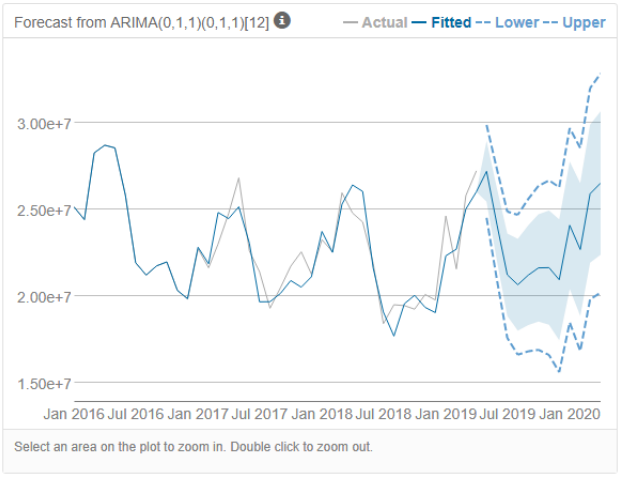
frequency as monthly. Under the "model customization" tab, I did not check "Completely user specified model" (which would have meant that for the options of the non-seasonal components and the seasonal components, I would have set 0, 1, 1 in accordance with my previously determined ARIMA(0, 1, 1), but 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.

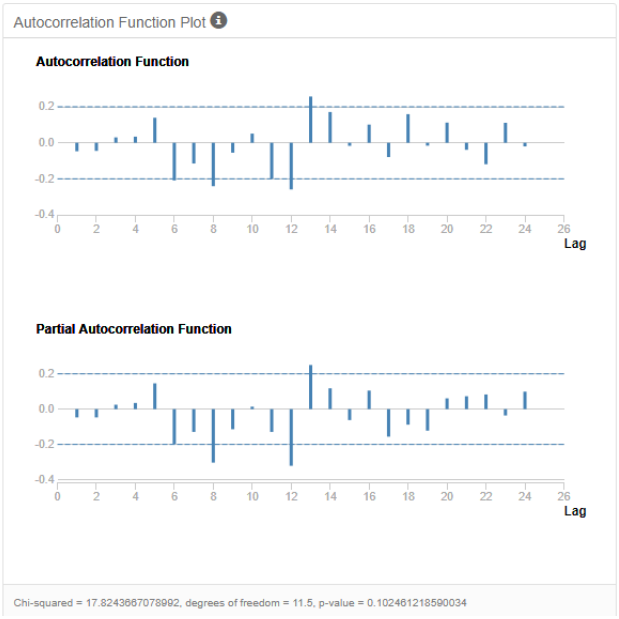
NB: The* "Customize the parameters used for automatic model creation" *option under the "Model customization" tab can be used to automatically determine the best ARIMA format without having to use the TS PLOT tool nor going through the steps taken above. But for this project, it was required that we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

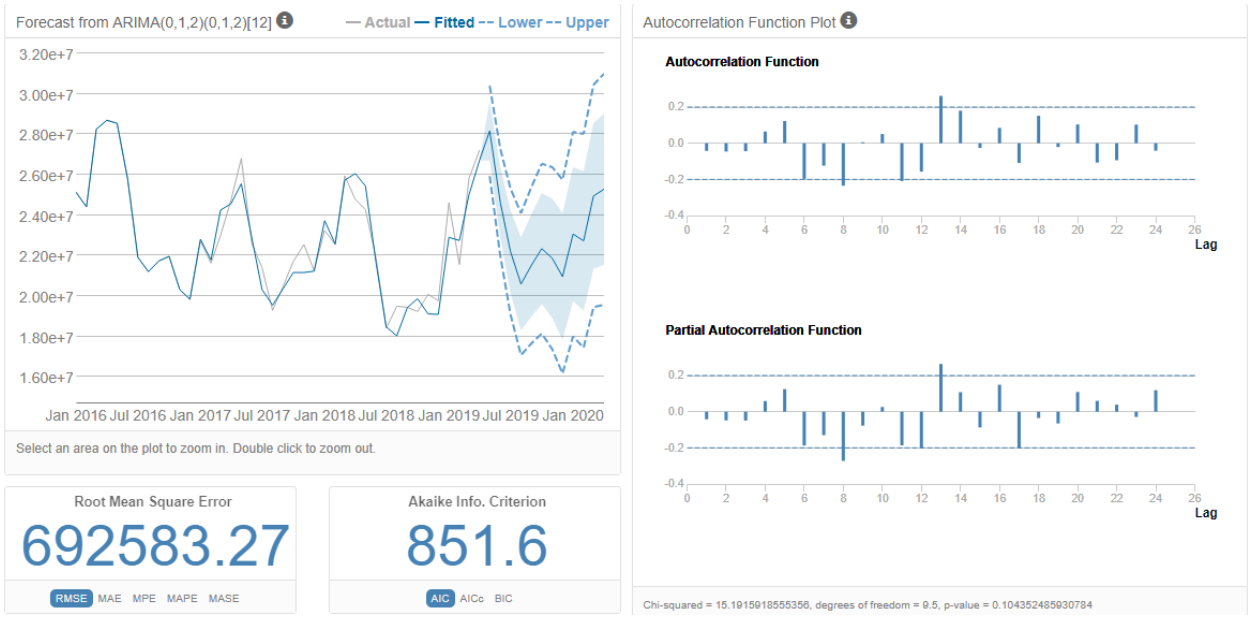
Now, under the "other options" tab, I set as same as I did for the ETS tool as stated above.

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:

i. 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.

ii. 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.

i. Residual plots

ii. Forecasting errors

iii. 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.

Accuracy Measures:

Model	ME	RMSE	MAE	MPE	MAPE	MASE
MNM	-21581.13	663707.2	553511.5	-0.0437	2.5135	0.3257
ARIMA_MA2	-1080158.09	1379900.3	1214618.2	-4.8109	5.4239	0.7147

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:

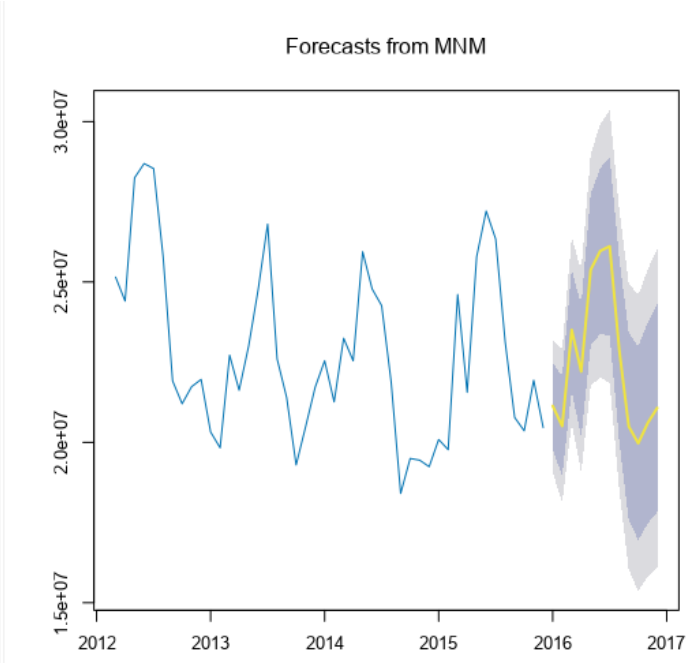
- Grouped by Year
- Grouped by Month
- Sum Produce.

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.

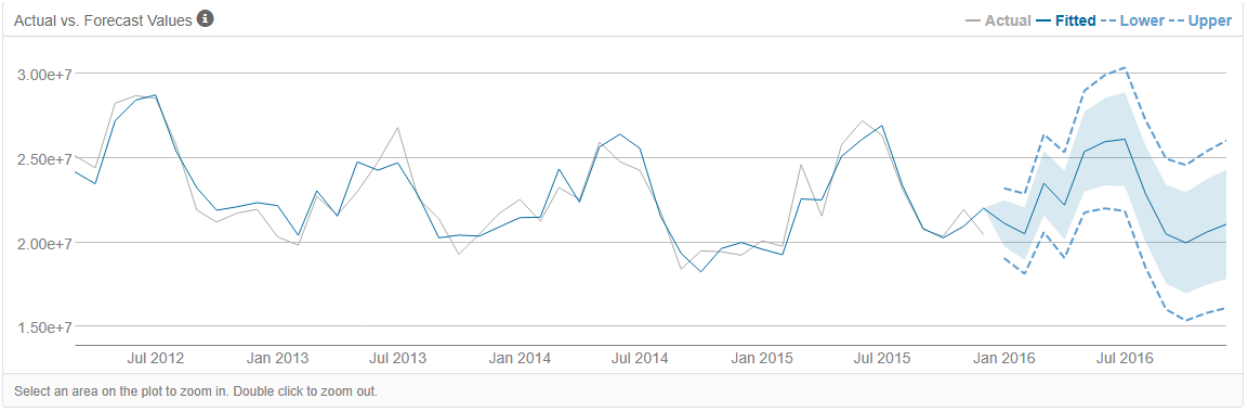
Then I connected a TS FORECAST tool (which can provide forecast for either an ETS or ARIMA model for a specified number of periods) to its output, and in the configuration panel I left the default generated "field name for the point forecast" as ***forecast*** and set the "number of periods into the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

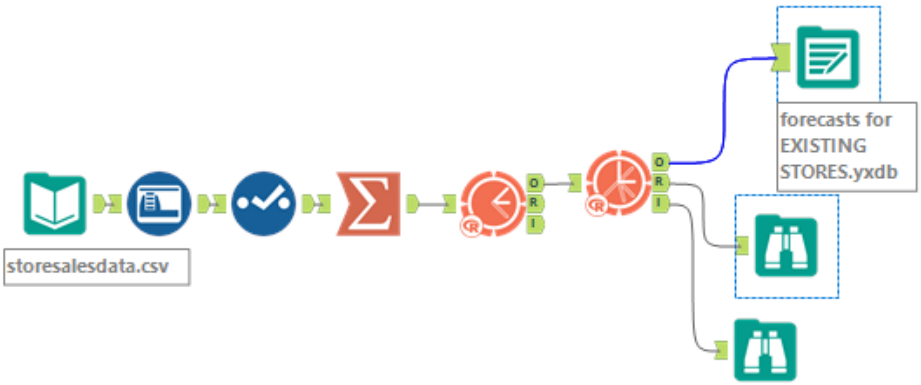


Period	Sub_Period	forecast	forecast_high_95	forecast_high_80	forecast_low_80	forecast_low_95
2016	1	21136641.781775	23208185.028684	22491151.105472	19782132.458079	19065098.534867
2016	2	20507039.12384	22880476.575432	22058946.457752	18955131.789929	18133601.672248
2016	3	23506565.982355	26405361.061884	25401986.205209	21611145.7595	20607770.902825
2016	4	22208405.755153	25340024.343149	24256061.087491	20160750.422815	19076787.167158
2016	5	25380147.771963	28988615.88472	27739598.24942	23020697.294506	21771679.659206
2016	6	25966799.465113	29918337.661734	28550571.413033	23383027.517192	22015261.268491
2016	7	26113792.565116	30368443.282705	28895759.137452	23331825.992781	21859141.847528
2016	8	22899285.769116	27261865.623116	25751823.410525	20046748.127707	18536705.915117
2016	9	20499583.908226	24962248.504876	23417563.445324	17581604.371129	16036919.311577
2016	10	19971242.820704	24578175.8382	22983554.407806	16958931.233603	15364309.803208
2016	11	20602665.916965	25388988.317688	23732273.917026	17473057.916904	15816343.516242
2016	12	21073222.081854	26036030.707662	24318228.221798	17828215.941909	16110413.456046

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.

Period	Sub_Period	forecast	forecast_high_95	forecast_high_80	forecast_low_80	forecast_low_95
2016	1	218778.521821	244341.813729	235493.459964	202063.583677	193215.229913
2016	2	211614.605364	238938.241544	229480.570759	193748.63997	184290.969185
2016	3	255219.376896	291035.062342	278637.994848	231800.758944	219403.69145
2016	4	233318.478259	268484.697642	256312.433179	210324.523339	198152.258876
2016	5	270368.562844	313748.400571	298733.114127	242004.011561	226988.725116
2016	6	274396.373282	320940.153575	304829.71541	243963.031153	227852.592988
2016	7	279885.352388	329798.144473	312521.572921	247249.131855	229972.560303
2016	8	239203.61528	283849.491753	268395.984803	210011.245757	194557.738807
2016	9	210409.490199	251356.469401	237183.280797	183635.699601	169462.510997
2016	10	207261.709567	249182.696323	234672.369472	179851.049663	165340.722812
2016	11	212498.193047	257046.833771	241626.983532	183369.402562	167949.552322
2016	12	218861.416999	266305.770396	249883.612229	187839.221769	171417.063602

Record	Period	Sub_Period	forecast	forecast_high_95	forecast_high_80	forecast_low_80	forecast_low_95
1	2,016	1	263,364.266251	287,677.405537	279,261.773532	247,466.75897	239,051.126965
2	2,016	2	254,026.065028	281,793.078484	272,181.939241	235,870.190815	226,259.051572
3	2,016	3	295,564.400276	332,215.437656	319,529.225499	271,599.575052	258,913.362895
4	2,016	4	283,905.868259	322,842.018653	309,364.84857	258,446.887947	244,969.717865
5	2,016	5	321,700.271616	369,679.406993	353,072.141988	290,328.401244	273,721.136239
6	2,016	6	327,352.512307	379,812.020233	361,653.940824	293,051.08379	274,893.00438
7	2,016	7	328,885.30505	385,010.355847	365,583.503223	292,187.106878	272,760.254253
8	2,016	8	296,870.657911	350,444.263106	331,900.555476	261,840.760347	243,297.052717
9	2,016	9	262,477.059661	312,287.748168	295,046.518254	229,907.601068	212,666.371154
10	2,016	10	256,705.846189	307,701.462618	290,050.087505	223,361.604873	205,710.22976
11	2,016	11	264,514.634746	319,312.309531	300,344.908396	228,684.361097	209,716.959961
12	2,016	12	256,259.005588	311,442.710121	292,341.690522	220,176.320655	201,075.301056

Period	Sub_Period	forecast	forecast_high_95	forecast_high_80	forecast_low_80	forecast_low_95
2016	1	254797.930238	281578.97294	272309.112828	237286.747648	228016.887535
2016	2	249384.577343	281203.221377	270189.670394	228579.484292	217565.93331
2016	3	294112.789046	337255.593874	322322.352847	265903.225245	250969.984218
2016	4	276042.727295	321201.631919	305570.547895	246514.906695	230883.822671
2016	5	313578.558255	369686.946237	350265.861198	276891.255312	257470.170274
2016	6	318847.58556	380407.097153	359099.18665	278595.98447	257288.073967
2016	7	319731.257134	385678.53476	362851.864222	276610.650045	253783.979508
2016	8	277319.40215	337960.924461	316970.762311	237668.041988	216677.879839
2016	9	245475.112875	302039.964894	282460.88161	208489.344141	188910.260856
2016	10	239751.368998	297683.215372	277630.967297	201871.770699	181819.522625
2016	11	252719.52873	316494.133963	294419.501701	211019.555759	188944.923496
2016	12	258031.784868	325803.844445	302345.552741	213718.016994	190259.72529

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

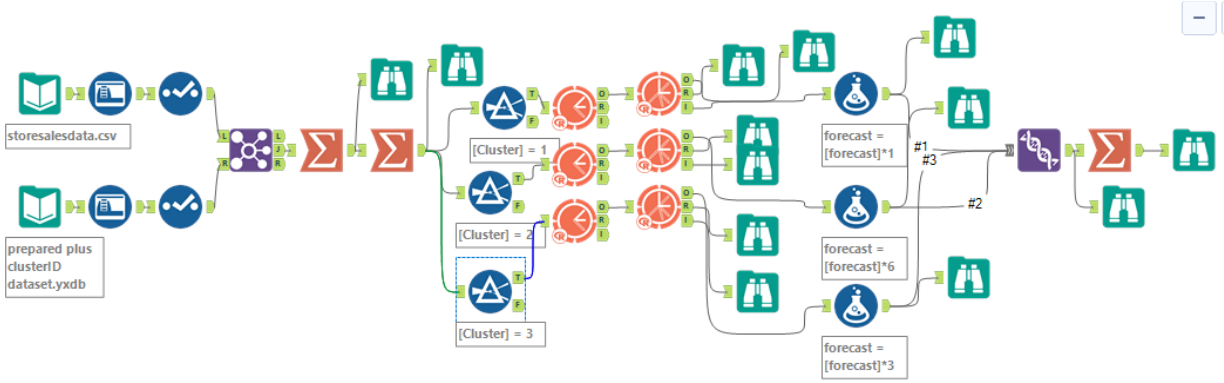
- Grouped by ***period***
- Grouped by ***Sub-period***
- Sum ***forecast***

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.

Record	Period	Sub_Period	Sum_forecast
1	2,016	1	2,563,357.910041
2	2,016	2	2,483,924.727562
3	2,016	3	2,910,944.145687
4	2,016	4	2,764,881.869697
5	2,016	5	3,141,305.867305
6	2,016	6	3,195,054.203804
7	2,016	7	3,212,390.95409
8	2,016	8	2,852,385.769198
9	2,016	9	2,521,697.18679
10	2,016	10	2,466,750.893696
11	2,016	11	2,557,744.587714
12	2,016	12	2,530,510.805133

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).

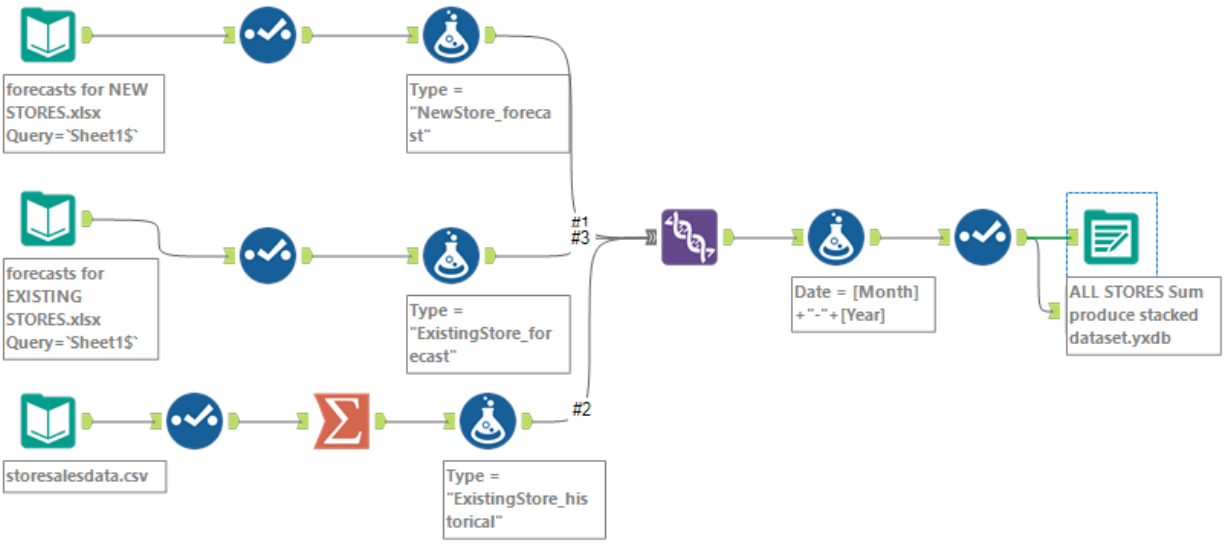
Then, I used a UNION tool to stack up the information in the three datasets whilst deselecting irrelevant fields. Next I used a FORMULA tool to create a new field ***Date*** to merge the ***Year*** and ***Month*** columns into a single date field, via using the formula: [Month]+"-"+[Year]. Finally, I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

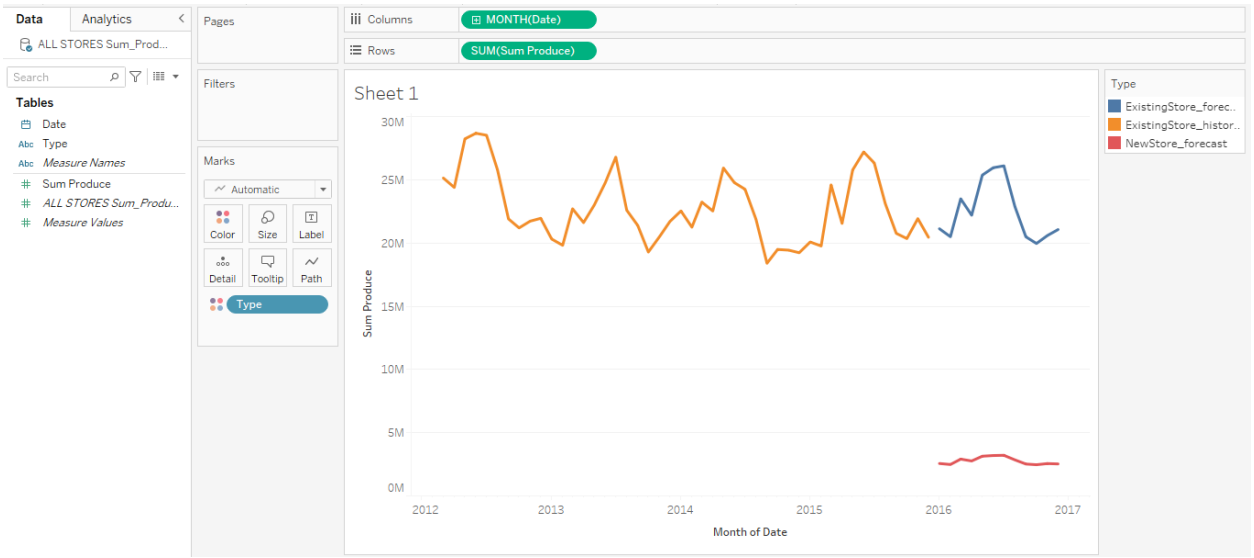
Record	Sum_Produce	Type	Date
1	2,563,357.910041	NewStore_forecast	1-2016
2	2,483,924.727562	NewStore_forecast	2-2016
3	2,910,944.145687	NewStore_forecast	3-2016
4	2,764,881.869697	NewStore_forecast	4-2016
5	3,141,305.867305	NewStore_forecast	5-2016
6	3,195,054.203804	NewStore_forecast	6-2016
7	3,212,390.95409	NewStore_forecast	7-2016
8	2,852,385.769198	NewStore_forecast	8-2016
9	2,521,697.18679	NewStore_forecast	9-2016
10	2,466,750.893696	NewStore_forecast	10-2016
11	2,557,744.587714	NewStore_forecast	11-2016
12	2,530,510.805133	NewStore_forecast	12-2016
13	21,136,641.781775	ExistingStore_forecast	1-2016
14	20,507,039.12384	ExistingStore_forecast	2-2016
15	23,506,565.982355	ExistingStore_forecast	3-2016
16	22,208,405.755153	ExistingStore_forecast	4-2016
17	25,380,147.771963	ExistingStore_forecast	5-2016
18	25,966,799.465113	ExistingStore_forecast	6-2016
19	26,113,792.565116	ExistingStore_forecast	7-2016
20	22,899,285.769116	ExistingStore_forecast	8-2016
21	20,499,583.908226	ExistingStore_forecast	9-2016
22	19,971,242.820704	ExistingStore_forecast	10-2016
23	20,602,665.916965	ExistingStore_forecast	11-2016
24	21,073,222.081854	ExistingStore_forecast	12-2016
25	25,151,525.84	ExistingStore_historical	03-2012
26	24,406,048.39	ExistingStore_historical	04-2012
27	28,249,539.01	ExistingStore_historical	05-2012
28	28,691,364.32	ExistingStore_historical	06-2012
29	28,535,707.45	ExistingStore_historical	07-2012
30	25,793,520.64	ExistingStore_historical	08-2012
31	21,915,641.66	ExistingStore_historical	09-2012
32	21,203,562.52	ExistingStore_historical	10-2012

33	21,736,158.96	ExistingStore_historical	11-2012
34	21,962,976.75	ExistingStore_historical	12-2012
35	20,322,683.64	ExistingStore_historical	01-2013
36	19,829,620.75	ExistingStore_historical	02-2013
37	22,717,069.85	ExistingStore_historical	03-2013
38	21,625,385.04	ExistingStore_historical	04-2013
39	23,000,152.4	ExistingStore_historical	05-2013
40	24,755,406.2	ExistingStore_historical	06-2013
41	26,803,105.56	ExistingStore_historical	07-2013
42	22,600,217.01	ExistingStore_historical	08-2013
43	21,401,265.74	ExistingStore_historical	09-2013
44	19,296,578.09	ExistingStore_historical	10-2013
45	20,489,773.49	ExistingStore_historical	11-2013
46	21,715,706.67	ExistingStore_historical	12-2013
47	22,544,458.38	ExistingStore_historical	01-2014
48	21,262,413.12	ExistingStore_historical	02-2014
49	23,247,168.62	ExistingStore_historical	03-2014
50	22,541,987.92	ExistingStore_historical	04-2014
51	25,943,046.75	ExistingStore_historical	05-2014
52	24,782,178.43	ExistingStore_historical	06-2014
53	24,263,117.59	ExistingStore_historical	07-2014
54	21,879,988.86	ExistingStore_historical	08-2014
55	18,407,263.58	ExistingStore_historical	09-2014
56	19,497,571.95	ExistingStore_historical	10-2014
57	19,444,753.17	ExistingStore_historical	11-2014
58	19,240,384.75	ExistingStore_historical	12-2014
59	20,088,529.29	ExistingStore_historical	01-2015
60	19,772,333.34	ExistingStore_historical	02-2015
61	24,608,406.71	ExistingStore_historical	03-2015
62	21,559,729.45	ExistingStore_historical	04-2015
63	25,792,074.59	ExistingStore_historical	05-2015
64	27,212,464.15	ExistingStore_historical	06-2015
65	26,338,477.15	ExistingStore_historical	07-2015
66	23,130,626.6	ExistingStore_historical	08-2015
67	20,774,415.93	ExistingStore_historical	09-2015
68	20,359,980.58	ExistingStore_historical	10-2015
69	21,936,906.81	ExistingStore_historical	11-2015
70	20,462,899.3	ExistingStore_historical	12-2015

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.