Alice Qi
20102432

November 19, 2021

CISC 251: Large Project

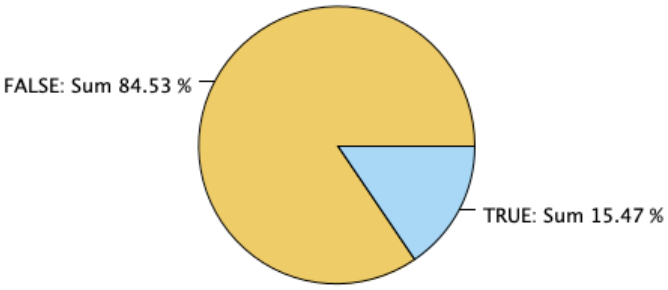
Goal of the Project

The goal of this project is to use various attributes collected about e-commerce shoppers to accurately determine whether a shopper will make a purchase or not. In order to make the model of use to an e-commerce site, the goal is also to determine which attributes/properties are the most important in leading to an eventual purchase. The goal of this project is to use the model and attribute ranking to suggest tangible changes to make to the e-commerce site that will drive up the number of customers who make a purchase, and ultimately lead to more profit for the online store.

Properties of the Dataset

Records: I am assuming each record represents one shopper or one site visit
Attributes: Quantitative and categorical data that is collected for each record
Target column: Revenue; a Boolean value indicating whether or not each record made a purchase

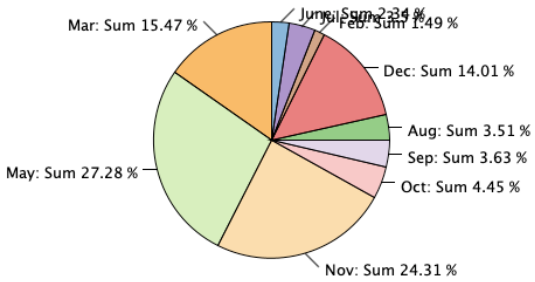
Breakdown of Revenue data



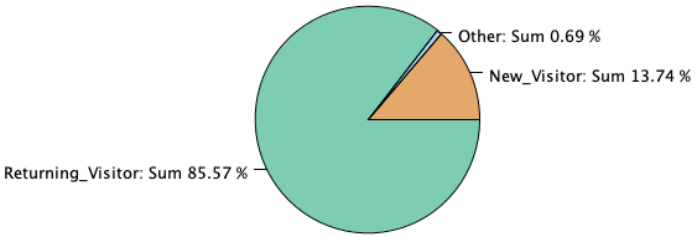
The vast majority of the data collected is for e-commerce shoppers who did not end up making a purchase. This is consistent with my expectations as many people will browse many stores multiple times before committing to purchasing. This will be important to consider in the event that a predictor classifies all values as FALSE, it will still output an accuracy of ~84%, which seemingly decent predictor just reflects the breakdown of the data of customers who did not make a purchase.

Breakdown of Categorical data

Month



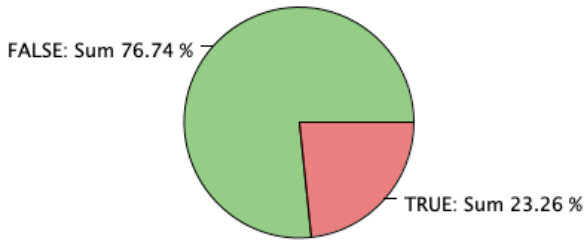
Visitor Type



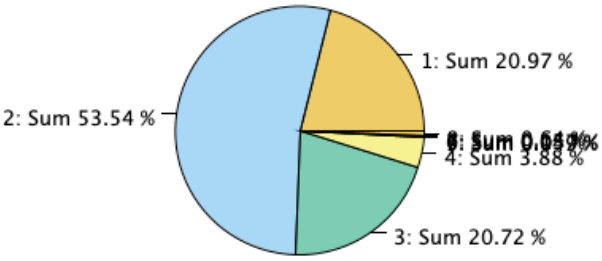
Alice Qi
20102432

November 19, 2021

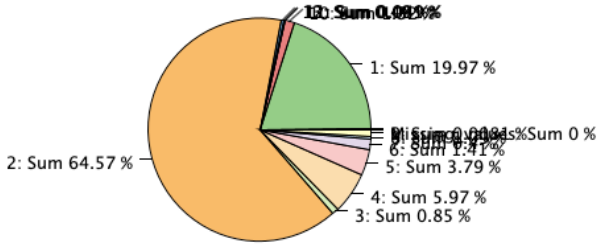
Weekend



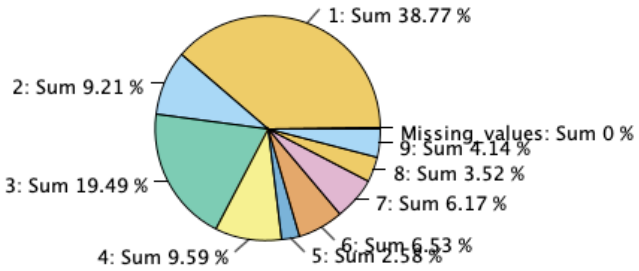
Operating System



Brower



Region

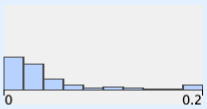
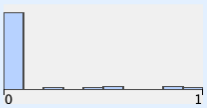


Properties of Numeric data

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. −∞	Histogram
Administrative	0.0	2.3152	?	27	3.3218	1.9604	4.7011	0	0	0	0 27
Administrative_Duration	0.0	80.8186	?	3,398.75	176.7791	5.6157	50.5567	0	0	0	0 3.399
Informational	0.0	0.5036	?	24	1.2702	4.0365	26.9323	0	0	0	0 24
Informational_Duration	0.0	34.4724	?	2,549.375	140.7493	7.5792	76.3169	0	0	0	0 2.549
ProductRelated	0.0	31.7315	?	705	44.4755	4.3415	31.2117	0	0	0	0 705
ProductRelated_Duration	0.0	1,194.7462	?	63,973.5222	1,913.6693	7.2632	137.1742	0	0	0	0 63.974

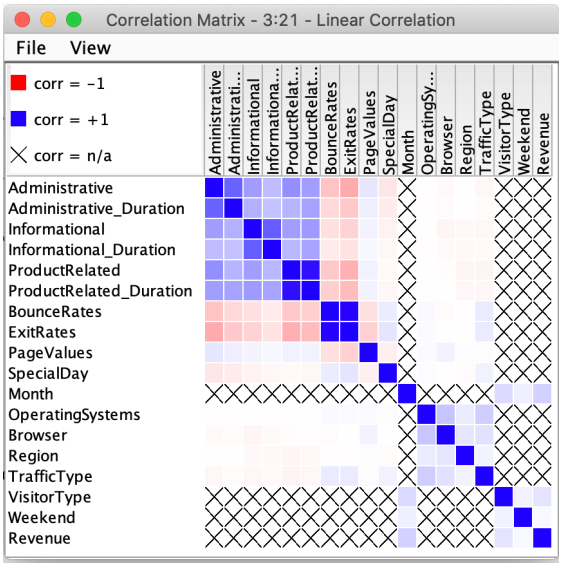
Alice Qi
20102432

November 19, 2021

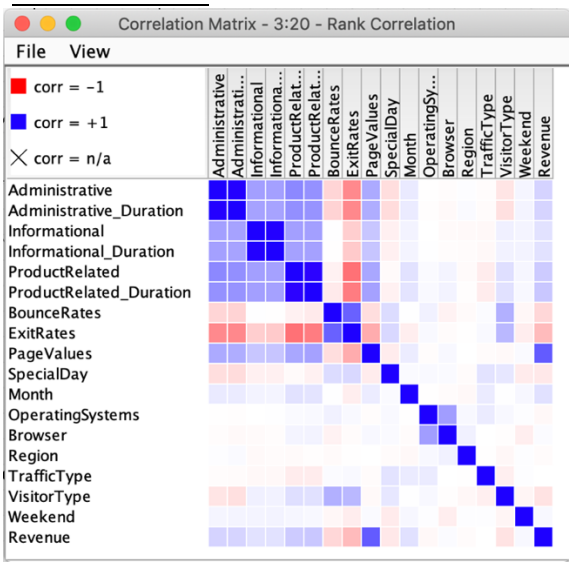
BounceRates	0.0	0.0222	?	0.2	0.0485	2.9479	7.7232	0	0	0	
ExitRates	0.0	0.0431	?	0.2	0.0486	2.1488	4.017	0	0	0	
PageValues	0.0	5.8893	?	361.7637	18.5684	6.383	65.6357	0	0	0	
SpecialDay	0.0	0.0614	?	1	0.1989	3.3027	9.9137	0	0	0	

Correlation

Linear Correlation



Rank Correlation



As seen in both linear and rank correlation, the initial 6 variables all have a strong positive correlation with one another. This demonstrates that each attribute collected are far from being independent from each other. Specifically Administrative, Informational, and Product Related are each highly correlated with their respective duration attribute. The correlation matrices also show how Exit Rates and Bounce Rates and negatively correlated with most of the other attributes as well as the target Revenue column.

Binning

I tried binning with some of the duration values, presumably the assumption is that there isn't too much meaning with the small differences in duration and I tried to bin the data by what would be considered an extremely short administrative duration (<5 minutes), a slightly longer administrative duration (5-15 minutes), 15-60 minutes, or a very long duration up to >60 minutes. The units for each duration variable were not specified. If the duration attributes were actually tracked using seconds for example, then the variance between numbers would be even smaller. However, binning rarely seemed to improve each predictor and so was later omitted from the Predictors section of the report.

Alice Qi
20102432

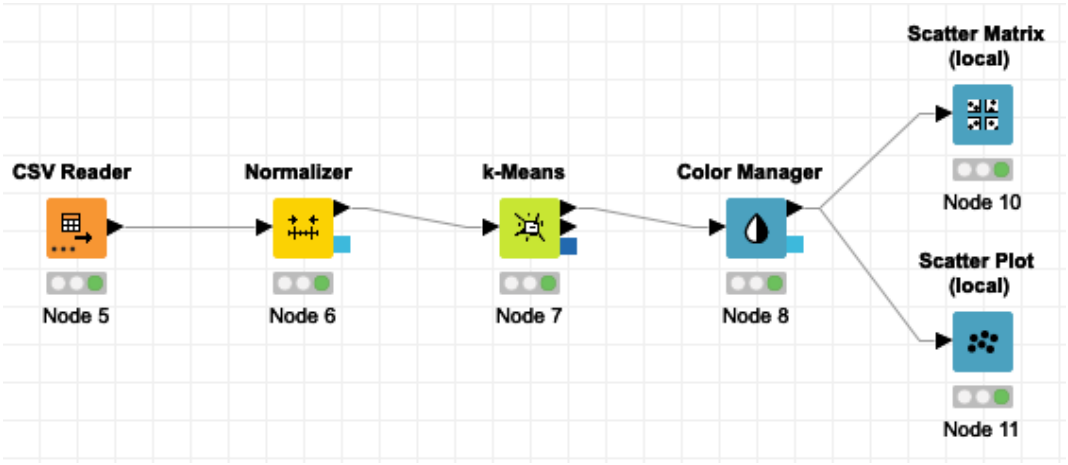
November 19, 2021

Clustering

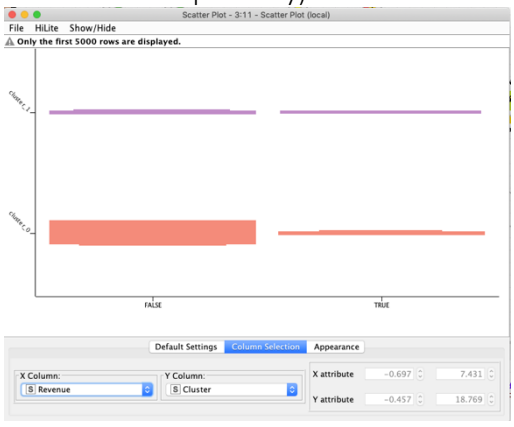
Goal: The ideal goal of clustering is to be able to segment the data by properties into 2 clusters, one that contains all the records of customers who made a purchase and the other cluster of records of customers who did not make a purchase.

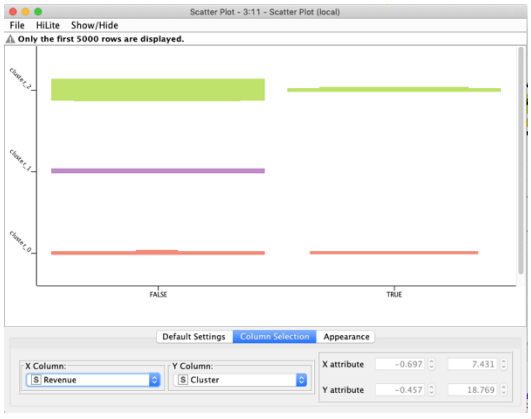
k-means

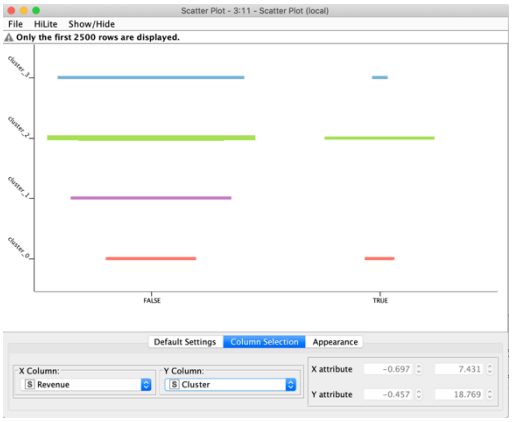
Workflow:



I first tried k-means clustering, using a k-value of 2, 3, and 4 (results are shown below for each k-value respectively).





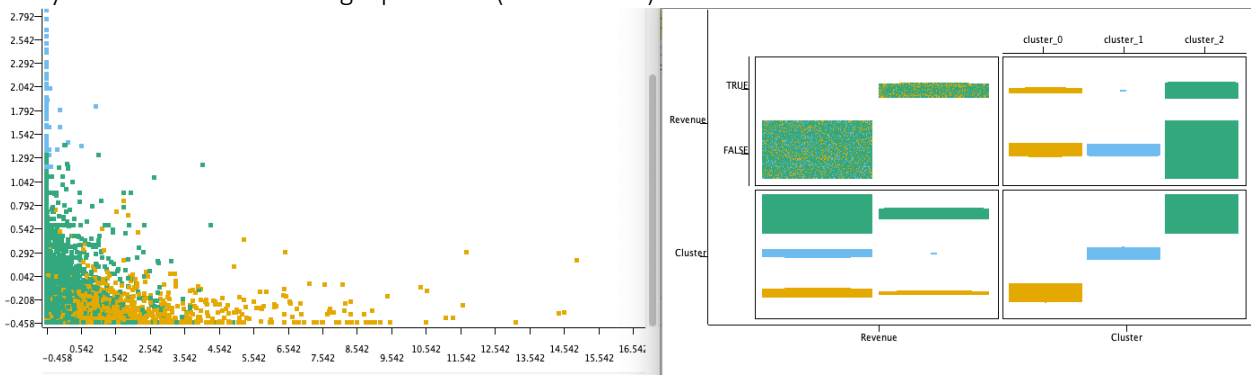


As illustrated with the scatterplots above, k-means clustering is not very successful at segmenting the data into clusters that divide values by the Revenue column. In each distinct column, they always contain more FALSE revenue values than TRUE revenue values. This is likely partially due to the fact that the dataset contains a majority of FALSE revenue values. No cluster can confidently be labelled as a cluster that represents shoppers who all purchased something. However, there was one interesting relationship that emerged when viewing the scatterplots for certain attributes.

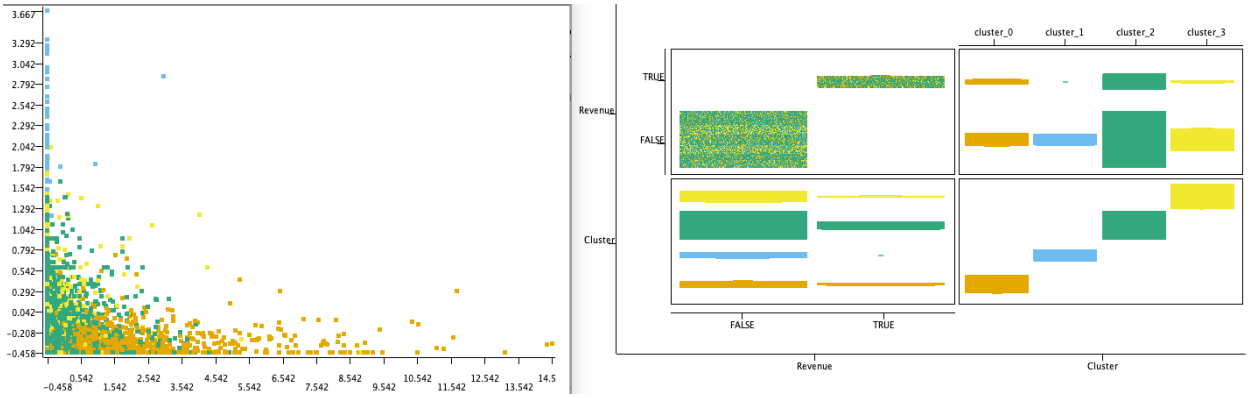
Bounce Rates vs Administrative Duration (each colour represents a cluster)

When k=3, it is seen that a pattern emerges between the length of time that a customer spends in administration (x-axis) and their bounce rate (y-axis). Cluster 0 represents the highest proportion of customers who do make a purchase (number of TRUE revenue values divided by number of FALSE revenue values per cluster), and the majority of cluster 0 (the brownish-yellow cluster) groups around relatively

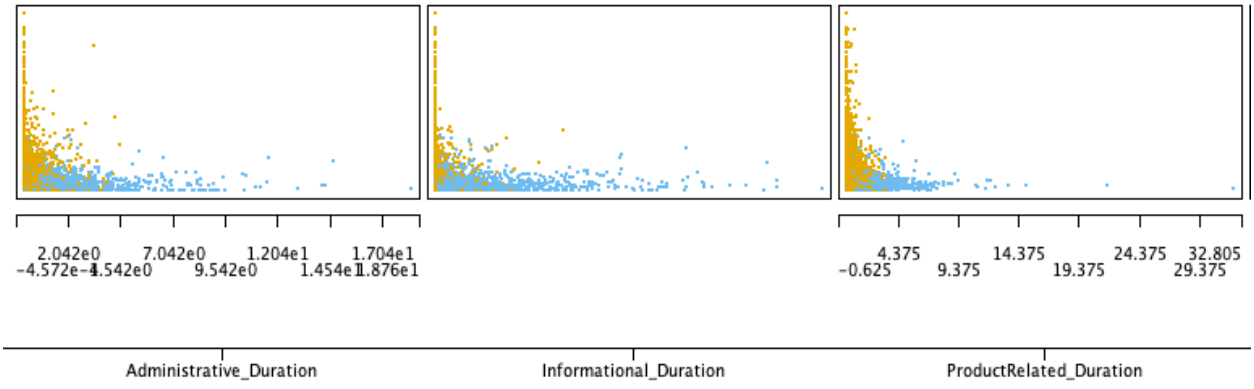
higher administrative duration values and has relatively low bounce rates when compared to the other 2 clusters. Conversely, cluster 1 (the blue cluster) contains mostly customers who did not make a purchase. Cluster 1 also tends to cluster around very low administrative duration values and have high bounce rates. This is consistent with what I would expect from customers who do not make a purchase, because it is likely that customers who do not commit a lot of time on the website (high administrative duration) are more likely to leave without making a purchase (bounce rate).



Consistent results can be seen when k=4:



These patterns also remain when comparing bounce rates (or exit rates) to any metric that measures the amount of time a potential customer spends on the website or time spent considering the product, including the informational duration and product related duration. Below the scatter matrix shows the patterns that emerge with k=2 clusters when comparing bounce rate (y-axis) to other duration metrics. As you can see, there appears to be some relationship between amount of time a potential customer spends on the site/collecting information and their likelihood of buying.

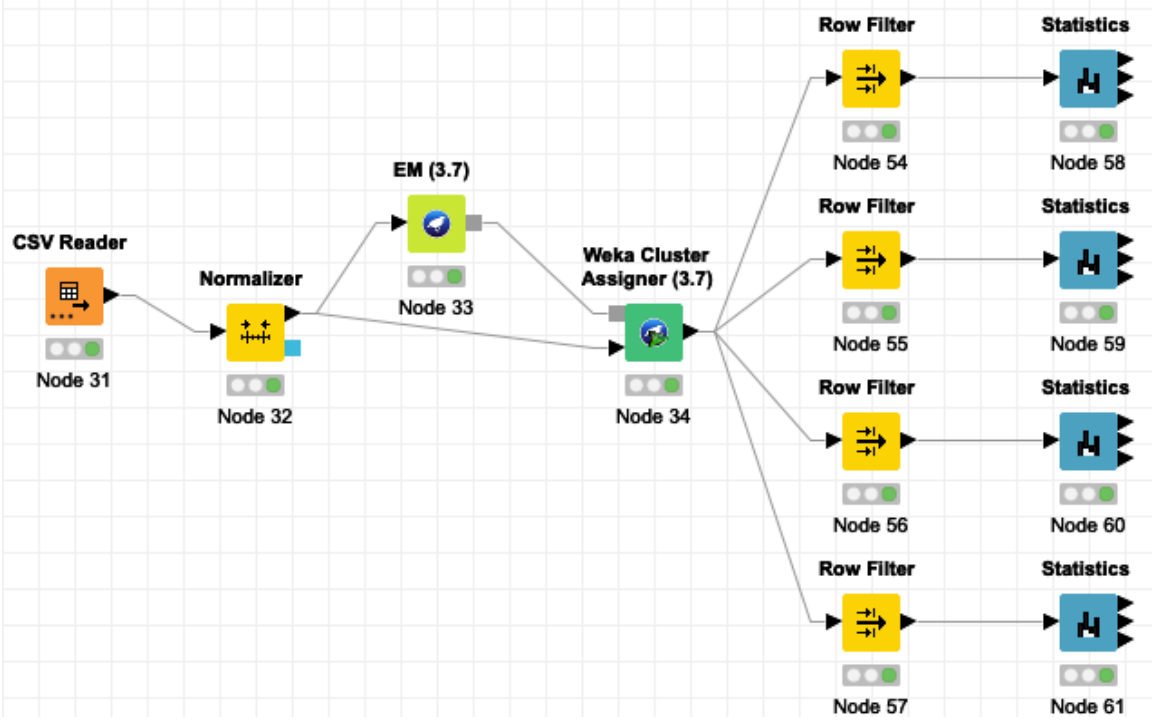


Alice Qi
20102432

November 19, 2021

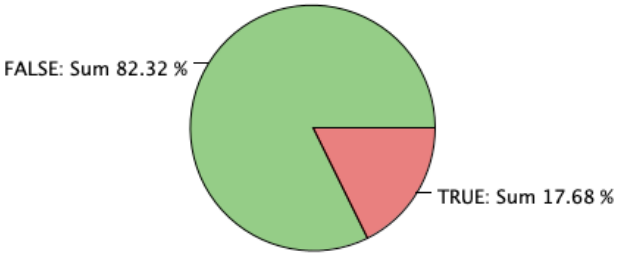
Expectation Maximization

Workflow:



Results with 4 clusters:

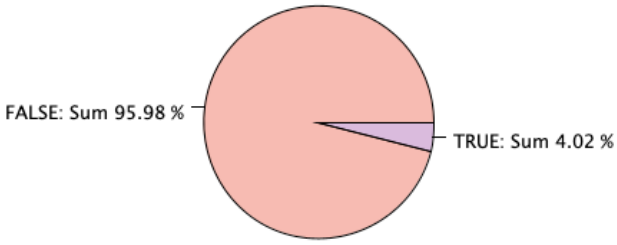
Cluster 0:



Cluster 0 has 4005 records

FALSE: 3297 records
TRUE: 708 records

Cluster 1:



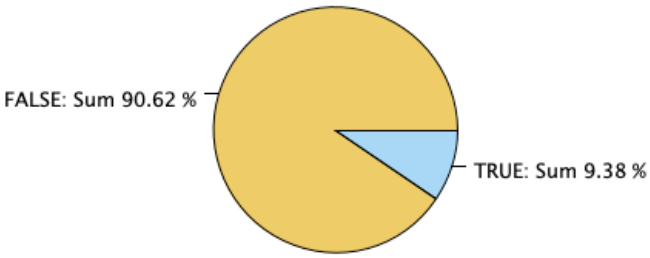
Cluster 1 has 1044 record

FALSE: 1002 records
TRUE: 42 records

Alice Qi
20102432

November 19, 2021

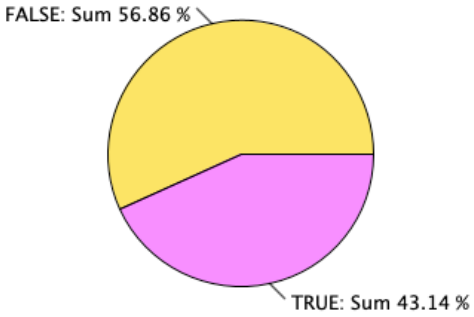
Cluster 2:



Cluster 2 has 5874 records

FALSE: 5323 records
TRUE: 551 records

Cluster 3:



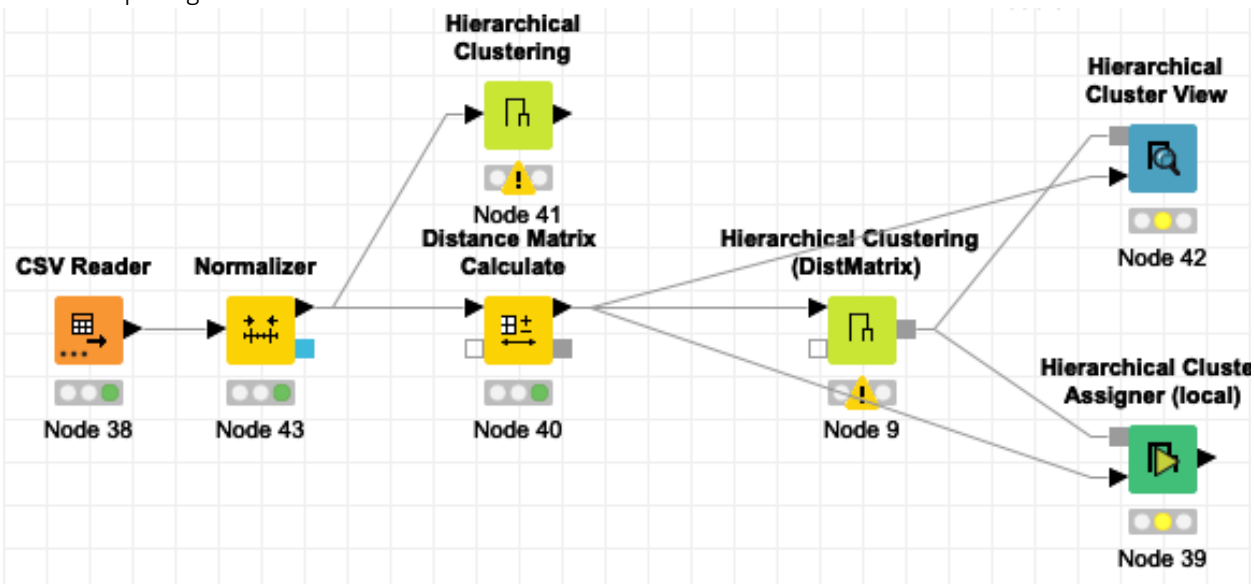
Cluster 3 has 1407 records

FALSE: 800
TRUE: 607

When using 2 clusters, it is clear that the majority of the TRUE results for revenue fall into Cluster 0. Comparably, with 4 clusters, most of the TRUE results for revenue fall into Cluster 3 and Cluster 0, whereas most of the results for FALSE fall into Cluster 1 and Cluster 2. This suggests that there may be underlying reasons for records in Cluster 0 and Cluster 3 to make a purchase, whereas records that fall into Cluster 1 or Cluster 2 contain properties that do not discern whether or not users will make a purchase.

Hierarchical Clustering

Hierarchical clustering was not used because the operation is too expensive to complete. Below is the workflow I set up, however even when letting it run for ~30 minutes, the algorithm was too expensive to finish computing.

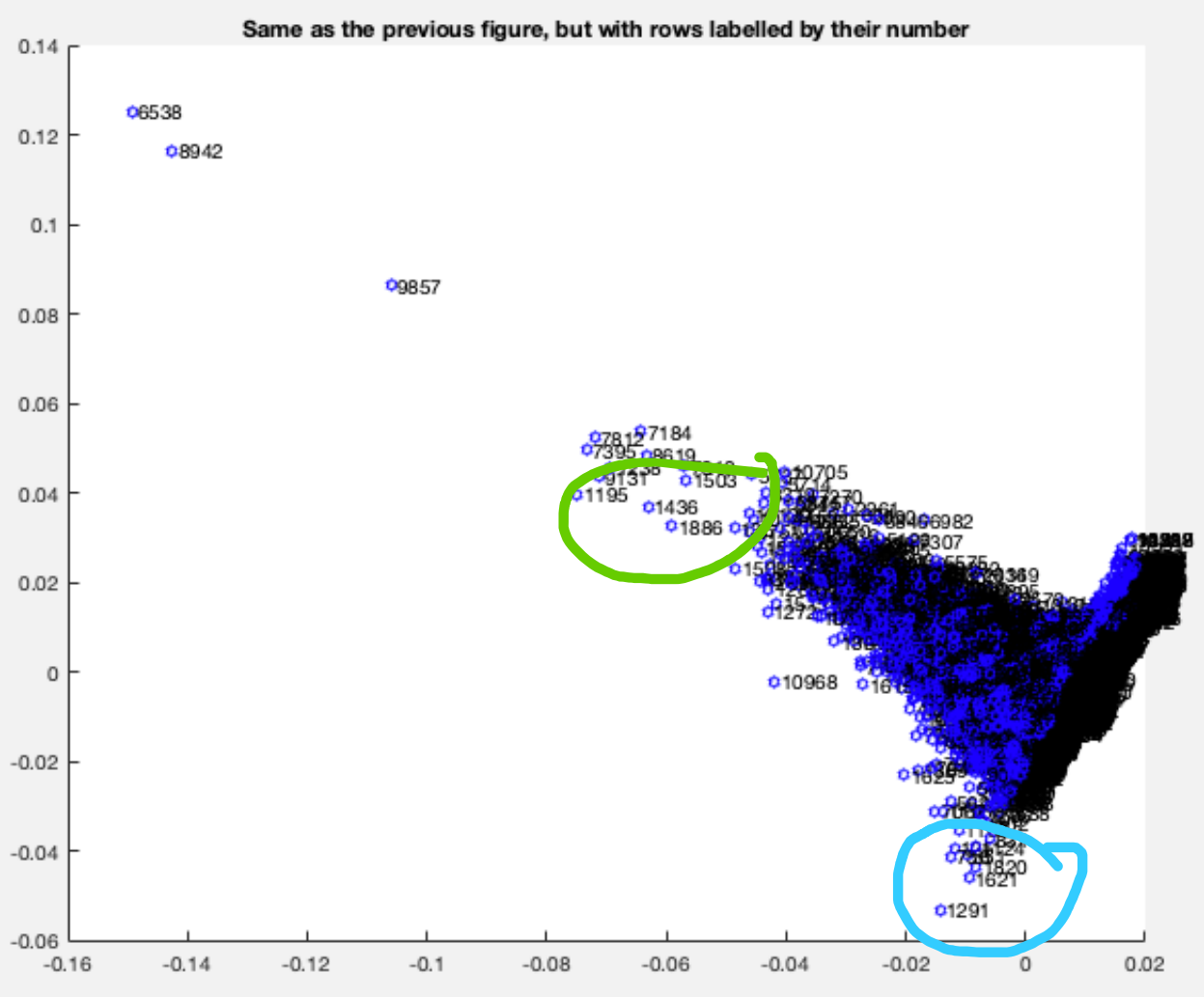


Alice Qi
20102432

November 19, 2021

Singular Value Decomposition

I first organized the data to display all the records that have TRUE in the revenue column at the top of the dataset in the first 1908 rows. When looking at the SVD that labels each row by number, a few clusters can be spotted as shown in the diagram below with the blue and green circles.



These 2 clusters are both far from the origin and contain a cluster of points that all represent TRUE revenue values seen as they both contain values from rows below row 1908. The SVD shows that there are potentially 2 unique reasons for customers to make a purchase that could be based on the two axis that are shown in the SVD, as these are the most “interesting” axis. However, the majority of points fall within the large center clustering of points surrounding the origin, which shows that this is a difficult prediction problem as there is no clear clustering between customers who make a purchase and customers who do not.

Alice Qi
20102432

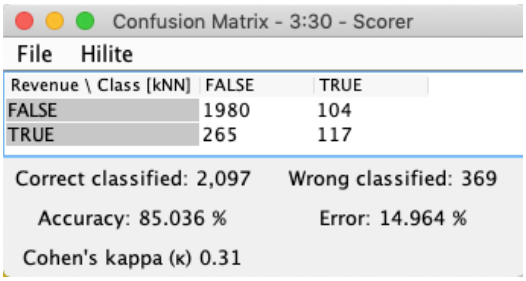
November 19, 2021

Predictors

K Nearest Neighbors

Workflow:

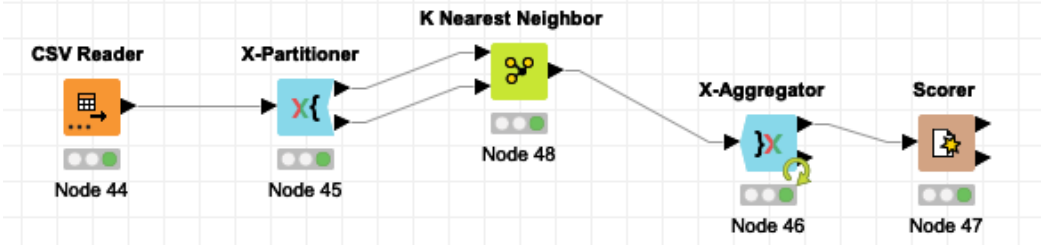
Results with k=4:



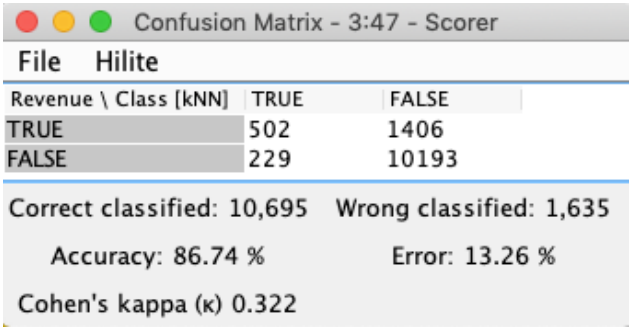
From the various clustering techniques that I employed, particularly the SVM, I don't think kNN would be a great technique because it seems like the data points aren't easily clustered between revenue TRUE and FALSE. When k is increased beyond k=4, the accuracy decreases. This is likely because the dataset contains mostly FALSE revenue values, so when k is increased to a value that is too large, the probability that the majority of the neighbors will be FALSE increases, simply due to absolute number of values.

I also applied cross validation with 100 validations and the accuracy increased by an immaterial amount to 86.74%.

Workflow using cross validation with 100 validations:



Results with k=4:



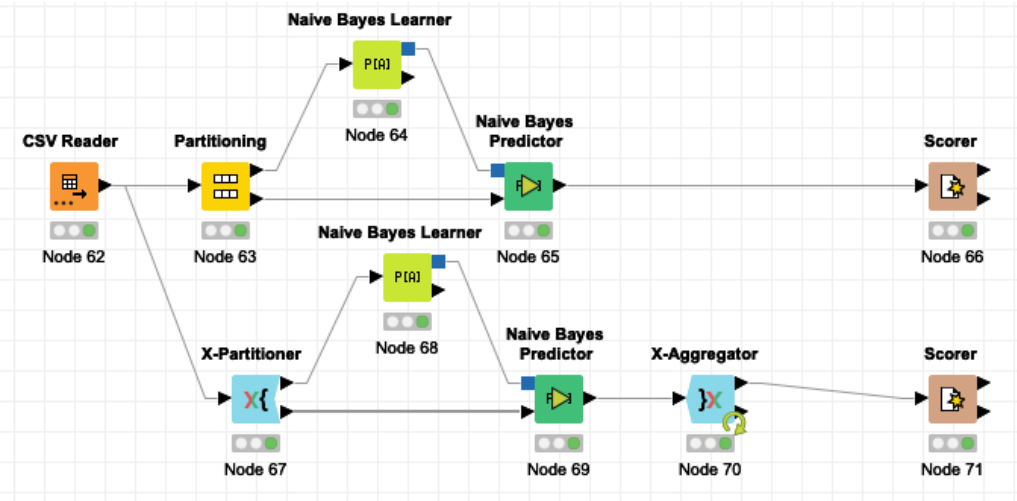
Alice Qi
20102432

November 19, 2021

Overall, I don't believe k-Nearest-Neighbors to be a great prediction technique, due to the fact that records for customers who did buy and who didn't buy are not easily separable as demonstrated from the clustering section of the project.

Naïve bayes

Workflow:



Results without using cross validation:

Confusion Matrix - 3:66 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	TRUE	FALSE
TRUE	242	140
FALSE	283	1801

Correct classified: 2,043 Wrong classified: 423

Accuracy: 82.847 % Error: 17.153 %

Cohen's kappa (κ) 0.432

Results using cross validation:

Confusion Matrix - 3:71 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	TRUE	FALSE
TRUE	1196	712
FALSE	1368	9054

Correct classified: 10,250 Wrong classified: 2,080

Accuracy: 83.131 % Error: 16.869 %

Cohen's kappa (κ) 0.435

Naïve Bayes predictor did not perform very well. When applying cross validation with 100 validations, the predictor still did not perform well, however improved slightly by <1% to an accuracy of 83.131%. I believe Naïve Bayes did not perform very well because it assumes that all predictors are independent when in reality this is far from the truth. For example, many attributes have a similar attribute that is closely related such as Administrative and Administrative Duration.

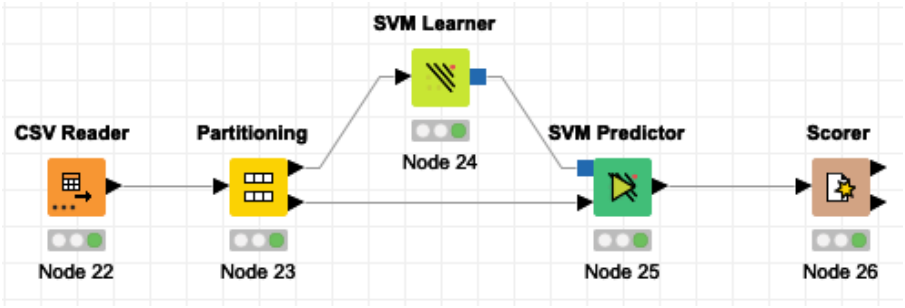
SVM

I wanted to try an SVM predictor because an SVM works well to classify data and the kernel trick can be used to transform the data to find complex relationships between datapoints.

Workflow:

Alice Qi
20102432

November 19, 2021



Results when using a HyperTangent kernel and C values 0.01, 0.1, 1, 10, 100, 1000:

Confusion Matrix - 3:26 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	FALSE	TRUE
FALSE	2079	5
TRUE	382	0

Correct classified: 2,079 Wrong classified: 387

Accuracy: 84.307 % Error: 15.693 %

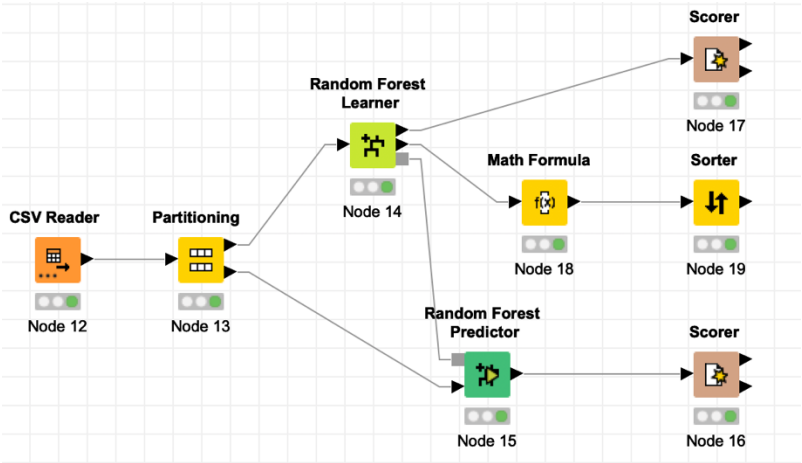
Cohen's kappa (κ) -0.004

When using a C value of 0.01 – 1000 all result in the same output and accuracy. This is interesting because the C parameter tells the SVM how much it should avoid misclassifying each training example. I am surprised that when testing smaller values of C because I thought the accuracy would drop significantly since the SVM would try to maximize the margin of the hyperplane at the expense of misclassifying data. What I believe is occurring is that the data is very difficult to separate and distinguish between customers who purchase and those who don't. The reasonable prediction accuracy of 83% is due to classifying most of the data as people who did not make a purchase, which represents about 83% of the data.

On KNIME, the SVM would only complete the execution when using a HyperTangent kernel. I chose a kappa of 0.5 and a delta of 1.0. This is likely because for other kernels, the computer struggled to find a way to split the data given the size and complexity of relationships between attributes.

Random Forests

Workflow:



Alice Qi
20102432

November 19, 2021

Results of Random Forest Predictor with all the attributes included:

Confusion Matrix - 3:16 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	FALSE	TRUE
FALSE	1987	97
TRUE	155	227

Correct classified: 2,214 Wrong classified: 252

Accuracy: 89.781 % Error: 10.219 %

Cohen's kappa (κ) 0.584

However, using the attribute statistics reported from the Random Forest Learner, I thought that the accuracy could be improved if we removed unimportant attributes. Removing the attributes that have little to no splitting influence could improve results because it increases the likelihood attributes with higher predictive power are chosen as split points on the tree.

Using the Math Formula and Sorter nodes, I added a column to the attributes statistics called "importance" that orders each attribute in descending order of importance, where importance is defined as (# splits chosen for level 0 / # of candidates at level 0) + (# splits chosen for level 1 / # of candidates at level 1) + (# splits chosen for level 2 / # of candidates at level 2). These results are displayed below:

Sorted Table - 3:19 - Sorter

File Edit Hilite Navigation View

Table "default" – Rows: 17 Spec – Columns: 7 Properties Flow Variables

Row ID	I #split...	I #split...	I #split...	I #cand...	I #cand...	I #cand...	D impor...
PageValues	129	204	286	129	230	458	2.511
ExitRates	95	102	156	119	242	474	1.549
Month	31	148	243	117	253	455	1.384
ProductRel...	68	91	145	110	257	439	1.303
BounceRates	67	78	125	115	231	443	1.202
ProductRel...	40	75	161	114	230	459	1.028
Administrat...	24	46	109	122	238	493	0.611
Administrat...	30	24	107	113	215	460	0.61
VisitorType	3	68	116	115	224	466	0.579
SpecialDay	7	41	88	125	229	453	0.429
Information...	4	29	93	114	231	465	0.361
Browser	2	24	68	126	248	459	0.261
OperatingS...	0	20	51	119	253	473	0.187
TrafficType	0	22	43	95	239	470	0.184
Informational	0	12	48	117	220	464	0.158
Region	0	0	16	132	210	455	0.035
Weekend	0	0	6	118	250	486	0.012

From the results of this table, I chose to remove the bottom 5 attributes because their "importance" or predictive power is very low.

Results with non-important attributes removed:

Alice Qi
20102432

November 19, 2021

Confusion Matrix - 3:16 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	FALSE	TRUE
FALSE	1990	94
TRUE	156	226

Correct classified: 2,216 Wrong classified: 250

Accuracy: 89.862 % Error: 10.138 %

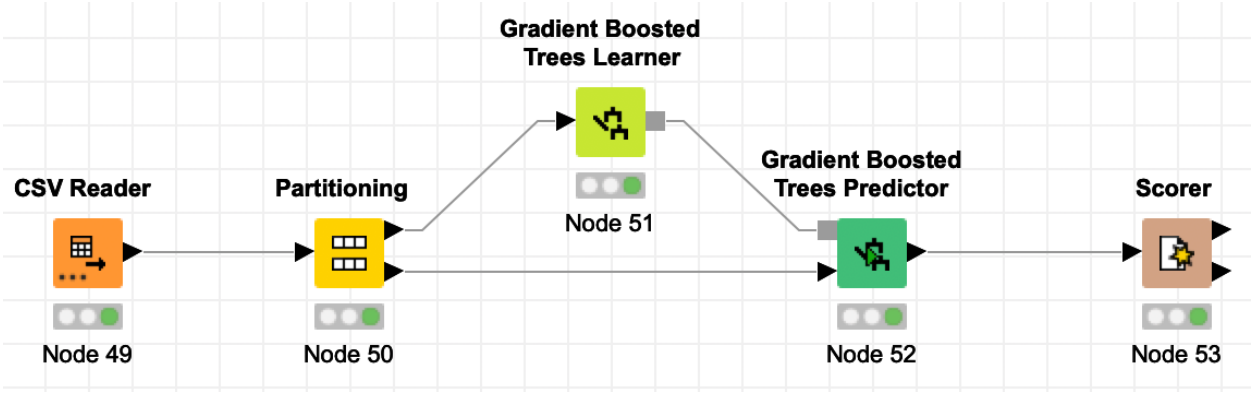
Cohen's kappa (κ) 0.585

With all the non-important attributes removed, the model performs with an accuracy of 89.862% which is immaterially higher than the accuracy without the 5 least important attributes removed. Results all hover around 89% whether using Gini Index, Information Gain, or the Information Gain Ratio.

Gradient Boosted Trees Learner

To try to improve on the accuracy of Random Forests, I wanted to try using a boosted technique by using a Gradient Boosted Trees Learner and Predictor. Initially, the accuracy was comparable to the accuracy of Random Forests, with an accuracy of 89.7%.

Workflow:



Results using all attributes:

Confusion Matrix - 3:53 - Scorer

File Hilite

Revenue \ Prediction (Revenue)	TRUE	FALSE
TRUE	221	161
FALSE	93	1991

Correct classified: 2,212 Wrong classified: 254

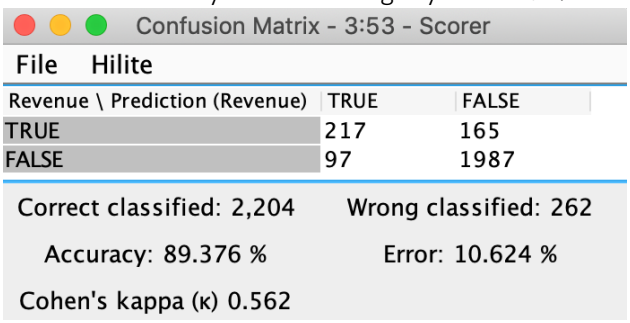
Accuracy: 89.7 % Error: 10.3 %

Cohen's kappa (κ) 0.576

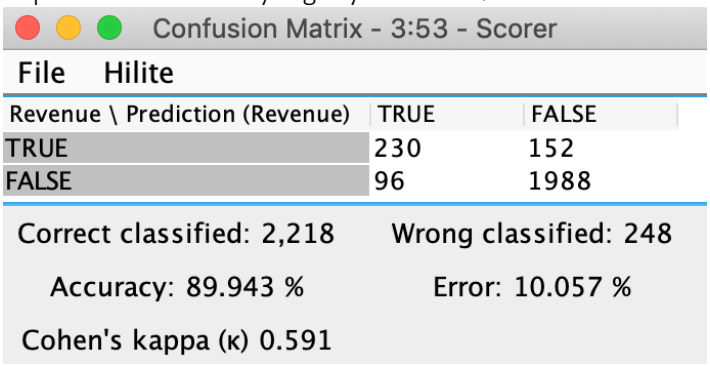
The accuracy was initially only ~85%, however when trying out different configurations for the Gradient Boosted Tree, the accuracy increased. One of the notable configuration choices I made was choosing to have data sampling choose a fraction of 100% of the data with replacement. This technique of bootstrapping ensures that the training data size is equal to the size of the data, while using ~2/3 of the

data. Another decision I found improved the accuracy of the model was to have no column sampling, which means each sample consists of all columns.

Another method I used to try to improve the accuracy of the predictor was to try removing the unimportant attributes beforehand. However, when removing the bottom 5 least predictive attributes, the accuracy of the model actually decreased slightly to 89.376%



Lastly, I tried to improve the accuracy by lowering the maximum tree depth from 8 to 4. This actually improved the accuracy slightly to 89.943%.



Summary of Predictors

After trying out various predictors include k Nearest Neighbors, Naïve Bayes, SVM, Random Forests, and Gradient Boosted Trees, it is seen that none of the predictors are especially accurate, as they all have prediction accuracies below 90%. Out of all the prediction techniques, Gradient Boosted Trees and Random Forests had the highest prediction accuracies of just over 89%. However, I would not recommend the e-commerce site rely on any of the predictors for classifying whether or not a shopper will buy.

Alice Qi
20102432

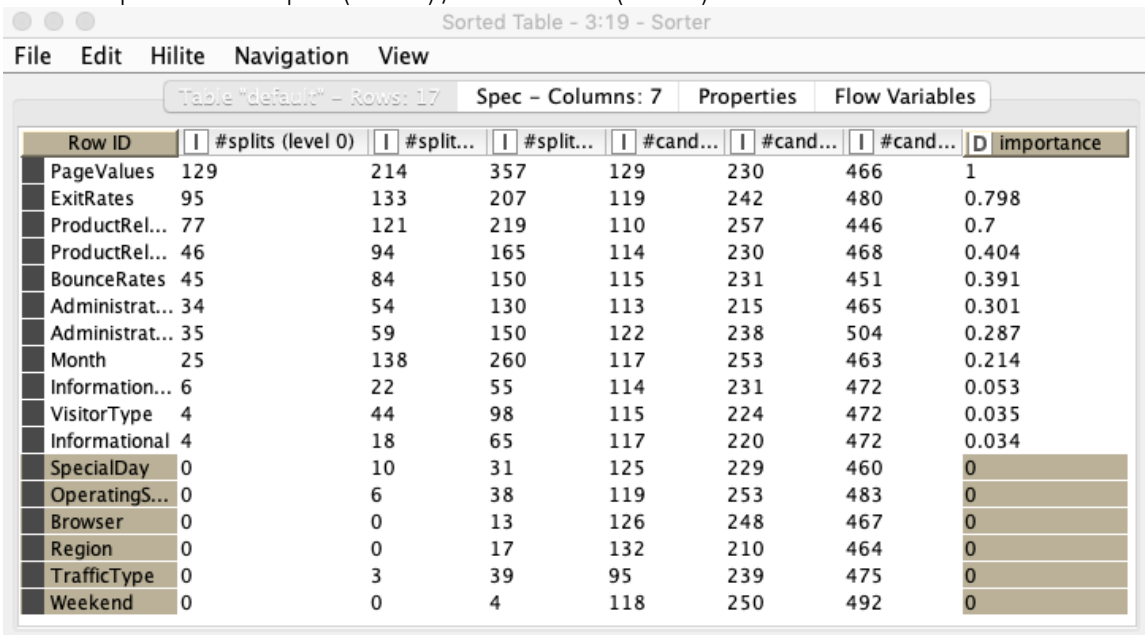
November 19, 2021

Attribute Selection and Ranking

I used 2 main techniques for attribute selection and ranking, these being using the Random Forest Attribute Statistics ranking and using rank correlation on all the columns.

Random Forests Attribute Ranking

Where importance = # splits (level 0) / # candidates (level 0)



Row ID	#splits (level 0)	#split...	#split...	#cand...	#cand...	#cand...	importance
PageValues	129	214	357	129	230	466	1
ExitRates	95	133	207	119	242	480	0.798
ProductRel...	77	121	219	110	257	446	0.7
ProductRel...	46	94	165	114	230	468	0.404
BounceRates	45	84	150	115	231	451	0.391
Administrat...	34	54	130	113	215	465	0.301
Administrat...	35	59	150	122	238	504	0.287
Month	25	138	260	117	253	463	0.214
Information...	6	22	55	114	231	472	0.053
VisitorType	4	44	98	115	224	472	0.035
Informational	4	18	65	117	220	472	0.034
SpecialDay	0	10	31	125	229	460	0
OperatingS...	0	6	38	119	253	483	0
Browser	0	0	13	126	248	467	0
Region	0	0	17	132	210	464	0
TrafficType	0	3	39	95	239	475	0
Weekend	0	0	4	118	250	492	0

From the attribute statistics, we can see that out of the 500 decision trees created, 129 of them used Page Values as their splitting variable at level 0. This suggests that Page Values has high predictive importance out of all of the attributes. Page Values refers to the dollar value of a given landing page that a user browses on. It is interesting to observe that Page Values are especially important in classifying whether or not a certain customer will buy. Furthermore, Page Values remains as the most important attribute when looking at level 1 and level 2 splits, and as a ratio of number of splits to number of candidates. Other important attributes appear to be Exit Rates and Product Related Duration. This also makes qualitative sense as number of users who exit a page should have a strong negative correlation with users who end up staying on a page and purchasing. Additionally, users who spend longer looking at and considering the product, have a higher chance that they are actually interested in the product and will purchase (Product Related Duration).

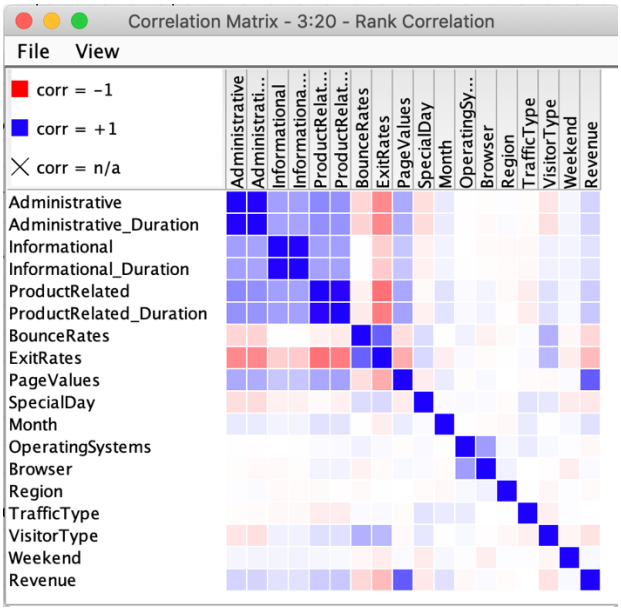
Other important information we can take away from the attribute statistics is which attributes have very low importance. This would include the 6 highlight attributes at the bottom on the importance ranking, all of which have an importance value calculated as 0. This is important information, because potentially some predictors could benefit from removing these unimportant attributes. Furthermore, the e-commerce site now knows not to spend a lot of resources on marketing techniques based on the bottom 6 attributes. For example, the e-commerce site should not offer discounts or promotions based on a customer's region or whether it is a weekend.

Alice Qi
20102432

November 19, 2021

Rank Correlation

I used rank correlation on all of the attributes in the dataset because the rank correlation node in KNIME automatically embeds the categorical data, so we can compare numeric values with the nominal Revenue column. I validated this by manually encoding the Revenue column to have FALSE values equal -1 and TRUE values equal to 1, which produced the same results as the rank correlation node below:



I wanted to use rank correlation to confirm the attribute rankings from the earlier analysis from the Random Forest Attribute Statistics. To begin, the matrix shows that Page Values has a strong positive correlation with Revenue. This is consistent with the ranking I found with the attribute statistics. The positive correlation suggests that as the value of a given landing page increases, the shopper is increasingly more likely to make a purchase. This is counter-intuitive at first, because it suggests users are not attracted to cheaper items on a page, and instead having a page filled with higher priced items would actually increase one's likeliness to purchase.

Another interesting insight from the rank correlation is seen with Exit Rates. The matrix shows that Exit Rates have a strong negative correlation with administrative duration and product related duration. This is consistent with the cluster analysis performed early in the report. Furthermore, Exit Rates have a relatively strong negative correlation with Revenue, also consistent with the results from the Random Forest Attribute Statistics. Lastly, Product Related Duration and Administrative Duration both have moderate positive correlations with revenue.

From both methods of attribute selection and ranking it is demonstrated that Page Values, Exit Rates, and Product Related Duration are among the most important attributes in classifying whether or not a customer will purchase. Conversely, Operating System, Brower, Region, Traffic Type, Special Day, and Weekend are unimportant attributes that can likely be omitted from most predictors without impacting the results (or even slightly improving the results).

Alice Qi
20102432

November 19, 2021

Actionable Conclusions

I recommend that the online shopping site make the following 3 changes in order to increase the proportion of users that end up making a purchase:

1. Increase the average page value of each page for all users
2. Redesign the UI/UX to encourage users to spend longer on product related pages
3. Create marketing campaigns that specifically target customers who have put in relatively a lot of time into administrative and informational activities

1. Increase the Average Page Value

From the analysis throughout the report, it is clear that Page Values is one of the most predictive attributes and has a strong positive correlation with a customer making a purchase. As such, I recommend that the online shopping site reorganize and design their page to ensure that each page that a user browses on has a high Page Value. This could be done by ensuring the standard landing page has high value popular items or by ensuring the personalized pages a user visits meets a minimum Page Value number, which would need to be determined from further analysis.

2. Redesign UI/UX to Encourage Users to Spend Longer on Product Related Page

The analysis throughout the report suggests that a longer duration spent on product related pages correlates negatively with exit and bounce rates, which subsequently correlates positively with likeliness to purchase. Therefore, I suggest that the online shop designs each product page so that customers are compelled to stay on the page or visit another relevant product page. Tactically, this could include having a recommendations engine for similar items that the user could easily click to view, or by having review or comments that would increase the duration they spend looking at a product.

3. Market Directly to Customers who have High Administrative and Informational Duration

Lastly, the analysis from the report shows that customers who commit a large portion of time to administrative and information activities are more likely to make a purchase. I recommend that the site tracks the time each customer is spending in administrative and informational activities as these customers are more likely to be interested in purchasing. The shop could then send promotions or bundle deals to these customers to encourage them to increase their basket size from what they would have originally ordered.

I believe that by implementing these 3 data driven recommendations, the online store will be able to increase their conversion ratio, revenue, and revenue per customer.