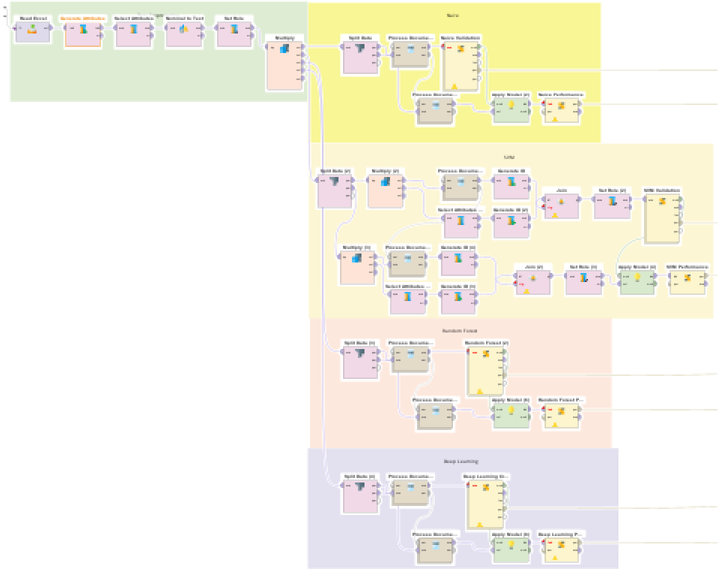
Redmi Dataset Report

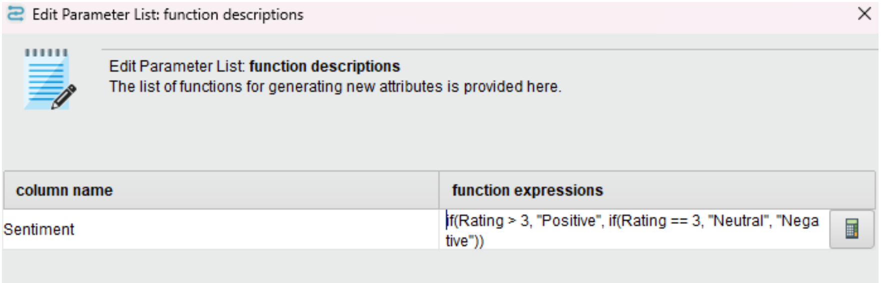
Created by: Solis, Alexander Jon S. Solis & Virata, Sean Maverick A.

This project uses the Redmi dataset with a goal to create a sentiment analysis machine learning model using the Altair AI Studio(RapidMiner) app. This dataset contains the following columns:

- Review Title - this contains the title of the customer review
- Customer Name - this contains the names of the customer
- Rating - this contains the numerical rating for the products(1 = lowest rating -> 5 = highest rating)
- Date - this contains the date of the customer review
- Category - this contains the category of the item
- Comments - this contains the customer review
- Useful - this contains the number of people who found the review useful



This dataset had gone through to this process as shown in the photo above this paragraph. First, we retrieved the data from the Redmi Dataset CSV file, then we proceeded with making a new feature which is the Sentiment column which has this function in it: if(Rating > 3, "Positive", if(Rating == 3, "Neutral", "Negative")). With this feature, we can determine what would be the positive and negative rating early in our analysis.

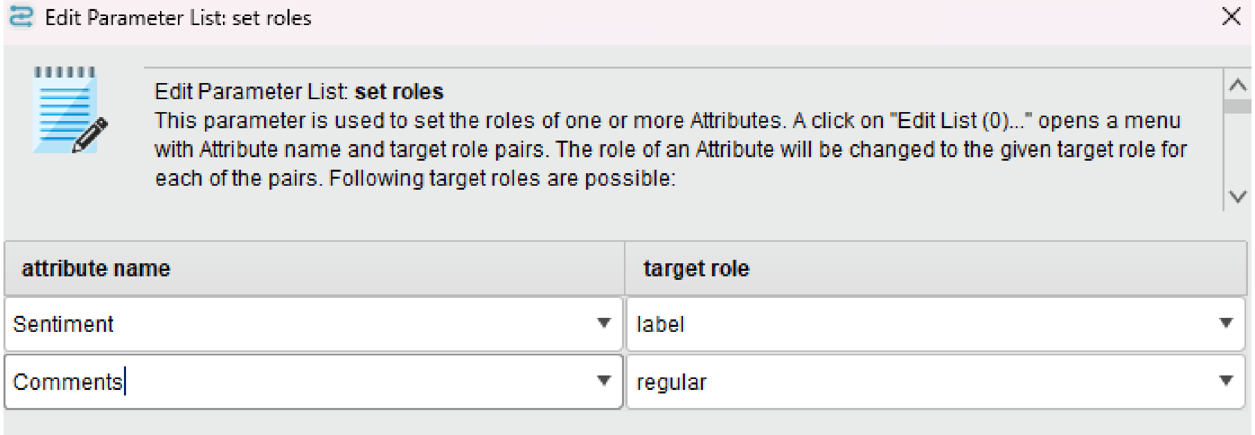


Edit Parameter List: function descriptions

Edit Parameter List: **function descriptions**
The list of functions for generating new attributes is provided here.

column name	function expressions
Sentiment	if(Rating > 3, "Positive", if(Rating == 3, "Neutral", "Negative"))

Next is that we selected three attributes that will be used for the sentiment analysis which are the: Comments, Review Title, and the Sentiment columns.What we did next is to convert the nominal values to test as it's necessary for the text analysis. After that, we selected the Sentiment as a target variable for the machine learning model by using the Set Role operator.

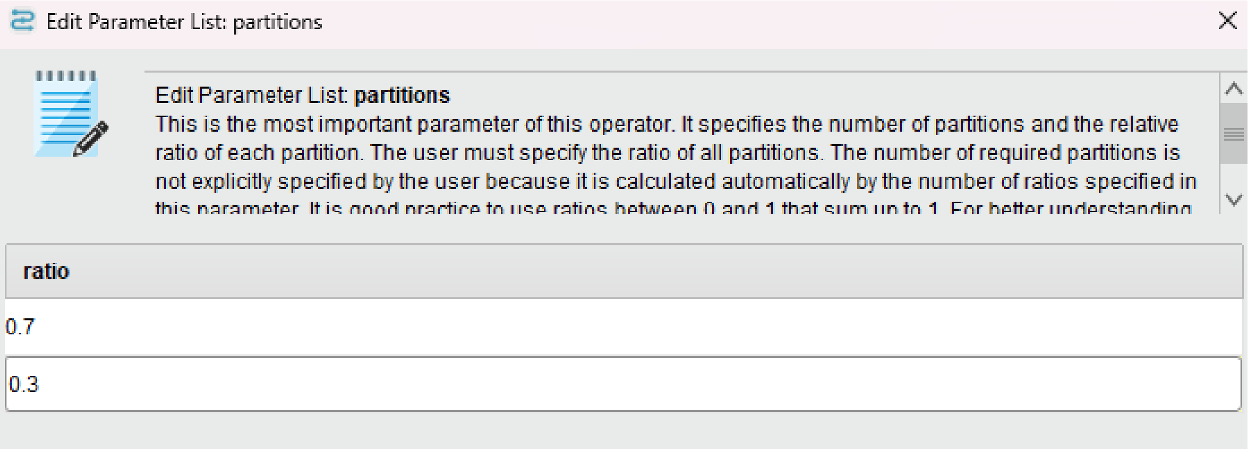


Edit Parameter List: set roles

Edit Parameter List: **set roles**
This parameter is used to set the roles of one or more Attributes. A click on "Edit List (0)..." opens a menu with Attribute name and target role pairs. The role of an Attribute will be changed to the given target role for each of the pairs. Following target roles are possible:

attribute name	target role
Sentiment	label
Comments	regular

The next thing we did was to split our data into a 70/30 ratio so the 70% of the dataset will be used to test the prediction of our model while the 30% of the data will be used to train the model before it is used on the test. This approach allows us to evaluate how well our model performs on unseen data which ensures that it generalizes well and does not just memorize the training set. Also, by keeping a larger portion for testing, we can better assess the model's real-world predictive capability, although this also means the model has less data to learn from, which can slightly affect its learning performance.

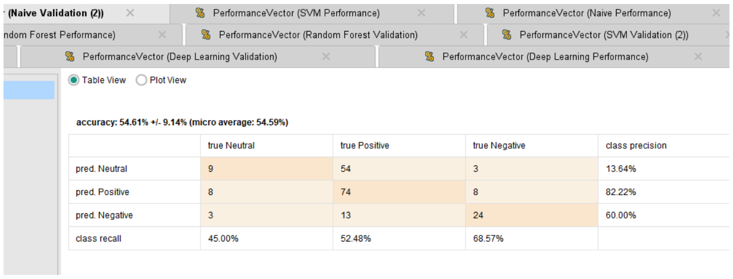


Edit Parameter List: partitions

Edit Parameter List: **partitions**
This is the most important parameter of this operator. It specifies the number of partitions and the relative ratio of each partition. The user must specify the ratio of all partitions. The number of required partitions is not explicitly specified by the user because it is calculated automatically by the number of ratios specified in this parameter. It is good practice to use ratios between 0 and 1 that sum up to 1. For better understanding

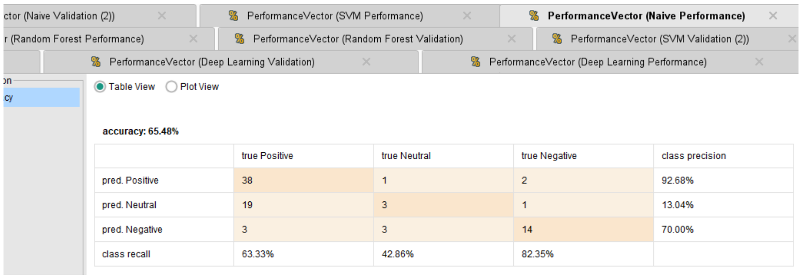
ratio
0.7
0.3

For our machine learning models, we trained Naive Bayes, Support Vector Machine (SVM), Random Forest, and Deep Learning on our dataset of Redmi 6 customer ratings for our sentiment analysis project. For text categorization, Naive Bayes is a simple, yet powerful algorithm that works well with high-dimensional data, like review texts. SVM, on the other hand, is well suited for sentiment analysis since it can handle big feature spaces and differentiate different categories between neutral, negative, and positive reviews. Given the noisy and diverse nature of customer reviews, Random Forest, an ensemble approach, mixes numerous decision trees to increase accuracy and lower the danger of overfitting. Lastly, when trained on a huge quantity of data, Deep Learning offers a stronger prediction ability by capturing complex trends and semantic associations in text. We were able to assess the precision, interpretability, and efficiency of these models to detect client sentiments by contrasting them with more sophisticated machine learning methods.


(Naive Validation (2))
PerformanceVector (SVM Performance)
PerformanceVector (Naive Performance)
ndom Forest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (SVM Validation (2))
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
● Table View
○ Plot View

accuracy: 54.61% +/- 9.14% (micro average: 54.59%)

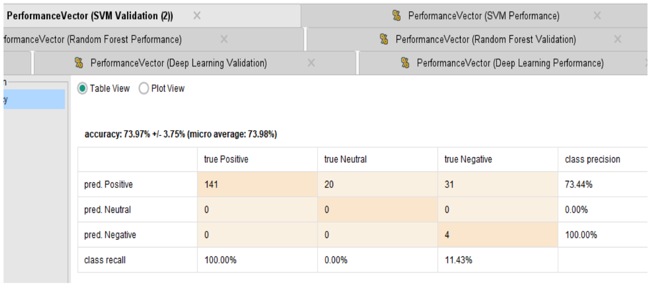
	true Neutral	true Positive	true Negative	class precision
pred. Neutral	9	54	3	13.64%
pred. Positive	8	74	8	82.22%
pred. Negative	3	13	24	60.00%
class recall	45.00%	52.48%	68.57%	


ctor (Naive Validation (2))
PerformanceVector (SVM Performance)
PerformanceVector (Naive Performance)
r (Random Forest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (SVM Validation (2))
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
on
cy
● Table View
○ Plot View

accuracy: 65.48%

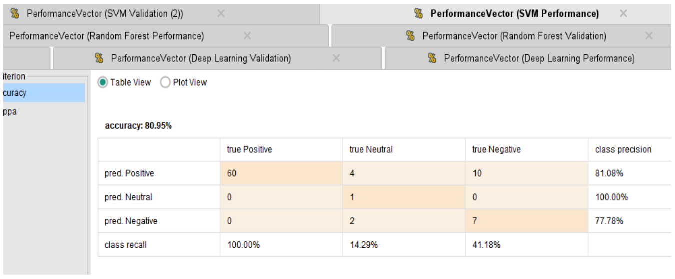
	true Positive	true Neutral	true Negative	class precision
pred. Positive	38	1	2	92.68%
pred. Neutral	19	3	1	13.04%
pred. Negative	3	3	14	70.00%
class recall	63.33%	42.86%	82.35%	

The Naive Bayes' training accuracy is 65% which is high. Similarly, its precision towards predicting the positive and negative reviews are high with 92.68% and 70% respectively while its precision towards predicting Neutral reviews are low with 13.04%. This indicates that the model performs well towards predicting Negative and Positive reviews but struggles to predict the Neutral reviews which is normal for sentiment analysis where it is hard to distinguish the neutral point for values. Meanwhile, for the validation or the test accuracy of the model, it got 54.61% for its accuracy score which is lower than the training accuracy. This may mean that the model might be overfitting the training data, as there is a slight difference between the training and test accuracy. Also, the values for predicting Negative reviews are much lower than the training set with a precision score of 60% while the Positive and Neutral reviews got 82.22% and 13.64% respectively which remains within the range of the model's precision score on the test data.


PerformanceVector (SVM Validation (2))
PerformanceVector (SVM Performance)
formanceVector (Random Forest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
n
y
● Table View
○ Plot View

accuracy: 73.97% +/- 3.75% (micro average: 73.98%)

	true Positive	true Neutral	true Negative	class precision
pred. Positive	141	20	31	73.44%
pred. Neutral	0	0	0	0.00%
pred. Negative	0	0	4	100.00%
class recall	100.00%	0.00%	11.43%	

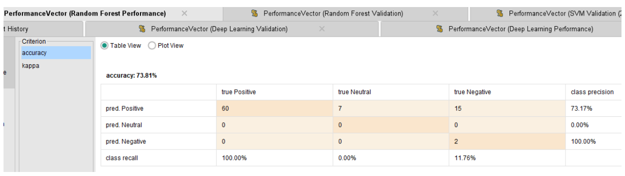

PerformanceVector (SVM Validation (2))
PerformanceVector (SVM Performance)
PerformanceVector (Random Forest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
iterion
curacy
ppa
● Table View
○ Plot View

accuracy: 80.95%

	true Positive	true Neutral	true Negative	class precision
pred. Positive	60	4	10	81.08%
pred. Neutral	0	1	0	100.00%
pred. Negative	0	2	7	77.78%
class recall	100.00%	14.29%	41.18%	

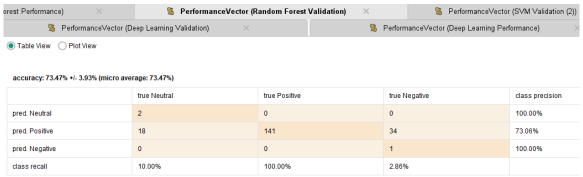
For the Support Vector Machine(SVM) model, the training accuracy was 80% which means that this model is much more efficient than the Naive Bayes algorithm. Meanwhile the prediction for Positive and Negative reviews precision scores got 81% and 77.78%. Even though the prediction for the Neutral review got the precision score of 100%, it's not enough to say that the model is effective for predicting true Neutral reviews as its recall score is 14.29% this means that the amount of Neutral reviews the model identified is low which is the same for the Negative reviews as its recall score is 11.43% which is low.

On the test accuracy of the Support Vector Machine(SVM) algorithm, the accuracy of the model is much lower than the training model with a score of 73.97% which may indicate again that the model has slightly overfitted on the dataset.


PerformanceVector (Random Forest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (SVM Validation (
t History
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
Criterion
accuracy
kappa
● Table View ○ Plot View

accuracy: 73.81%

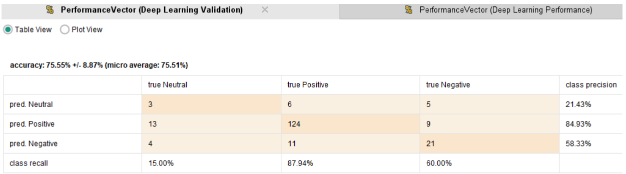
	true Positive	true Neutral	true Negative	class precision
pred. Positive	60	7	15	73.17%
pred. Neutral	0	0	0	0.00%
pred. Negative	0	0	2	100.00%
class recall	100.00%	0.00%	11.76%	


orest Performance)
PerformanceVector (Random Forest Validation)
PerformanceVector (SVM Validation (2))
PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
● Table View ○ Plot View

accuracy: 73.47% +/- 3.93% (micro average: 73.47%)

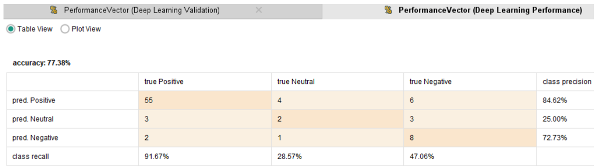
	true Neutral	true Positive	true Negative	class precision
pred. Neutral	2	0	0	100.00%
pred. Positive	18	141	34	73.06%
pred. Negative	0	0	1	100.00%
class recall	10.00%	100.00%	2.86%	

The Random Forest model delivered 73.47% accuracy during training and a 73.81% when the model was applied to determine its performance. The 100% class precision it had achieved when predicting the negative sentiment during the test set indicated a high likelihood of an overfitting to the dataset. Despite its strong performance, the dataset overfit meant that it could not consistently produce the same performance using a different dataset.


PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
● Table View ○ Plot View

accuracy: 75.55% +/- 8.87% (micro average: 75.51%)

	true Neutral	true Positive	true Negative	class precision
pred. Neutral	3	6	5	21.43%
pred. Positive	13	124	9	84.93%
pred. Negative	4	11	21	58.33%
class recall	15.00%	87.94%	60.00%	


PerformanceVector (Deep Learning Validation)
PerformanceVector (Deep Learning Performance)
● Table View ○ Plot View

accuracy: 77.38%

	true Positive	true Neutral	true Negative	class precision
pred. Positive	55	4	6	84.62%
pred. Neutral	3	2	3	25.00%
pred. Negative	2	1	8	72.73%
class recall	91.67%	28.57%	47.06%	

Lastly, the Deep learning model has achieved an accuracy of 75.55% on its training set while having a higher accuracy on its test set at 77.38%. It is interesting to note that the class precision when predicting a neutral is worse than when identifying positive or negative sentiment. During the training set, the class precision when predicting a neutral sentiment is at 21.43% compared to 84.93% and 53.33% when predicting positive or negative sentiments, respectively.

This project has achieved its goal in determining the customer sentiment of the Redmi dataset. After the data has been preprocessed, with the used of the Altair AI Studio, the dataset has underwent a data analysis process using different algorithm models to determine the customer's sentiment. The SVM model has achieved the best accuracy at 80.95% while Naive Bayes' is the worst performer at 65% in this metric. It is important to note that the models are prone to overfitting with the given dataset.