BDA 622 Marketing Analytics – Final Project
Text Mining Analysis of Movie Reviews

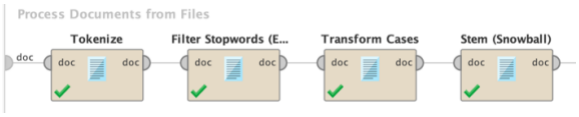
Jairo Onate
MUID 11062465
12/12/2024

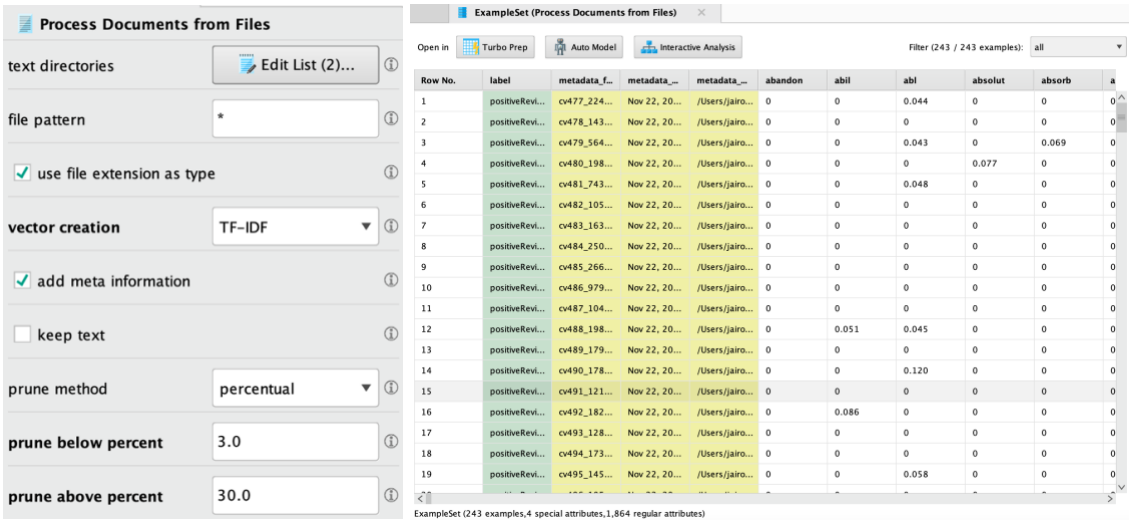
The project Text Mining Analysis of Movie Reviews aims to leverage text mining techniques and natural language processing (NLP), to analyze the data obtained from movie reviews, reveling patterns through textual data. The tool RapidMiner has been used to process text data, to automatically process a large number of documents (movie reviews) with a positive or negative label, in order to perform future predictions whether an incoming and unknown review will be classified as positive or negative. Being able to perform the sentiment analysis through words, will allow us to understand the customers perception towards the movie, providing insights to conduct further modifications in the marketing strategy.

The dataset that has been used to perform the analysis was decoded from txt (text) files, which has previously divided into a negative and positive label. Once the data has been processed and converted into data points, we encountered that the dataset used contains 243 examples (data points), with 10,761 regular attributes. Additionally, the set contains 4 special attributes, where label is the most important as it summarizes whether the data point was classified positively or negatively.

To develop a model that classifies text data, the following steps have been performed:

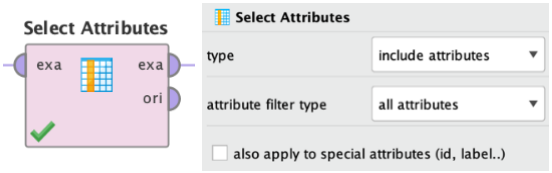
1. **Text processing:**
The extension Text Processing has been added on RapidMiner, where the operator Process Documents from Files was used to manipulate the initial text data. First, the operator Tokenize was applied to convert the documents into data points. After Tokenizing and obtaining a list of words, the method TF-IDF was selected to assign weights to the attributes (words) and create the vector. To prune the keywords and reduce the dimension of the data, the percental threshold selected was 3% and 30%, where the words which their frequency is below the minimum of above the maximum threshold have been omitted, keeping 1,864 attributes after preprocessing. The following image shows the workflow of converting documents into data points, converting the words to low case and stemming the words to use the root as the main attribute, the parameters for creating the vector and the final attributes kept after pruning.





2. Data partition:

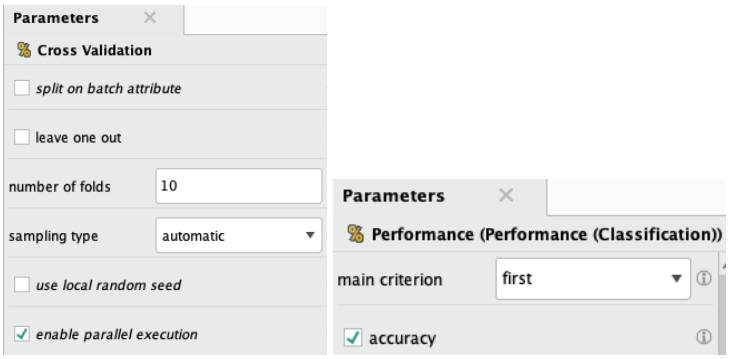
The operator Select Attributes was added to guarantee that the algorithm knows the attribute type and which attribute to include when modeling. These parameters have been specified to include attributes and select all of them.

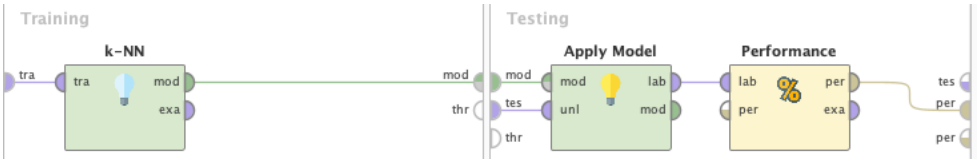


Part of the process of constructing the model consists of portioning the data, where the operator Set Role references the variables previously classified, giving the to the algorithm the path (directory) of the positive and negative data points. The attribute's selected type is label and the targe role (our variable of interest to predict) was defined as target, since the purpose is to build a classification model.

3. Model construction:

The model was built under the Cross Validation operator, where the number of folds meaning the number of subsets to train the model was selected was 10. The sampling type has been specified as automatic, using by default stratified sampling. After testing K-Nearest Neighbors, Random Forest and Decision Tree, the K-NN model with k = 23 had the best performance. The model operator has been placed under the training side, the cosine similarity numerical measure has been chosen to calculate the distance between the attribute vectors and the metric to measure the performance was accuracy. The following images show the operators, the process flow and each section parameters for the K-NN with k = 23 model.





4. Model evaluation

The model K Nearest Neighbor had the best performance, with an accuracy of 71.16% when classifying the reviews with a positive or negative review. This algorithm reported a Sensitivity (prediction of true positive reviews) of 67.90%, while the Specificity (prediction of true negative reviews) was 74.07%. In the appendix section, a screenshot summarizing the models compared with their corresponding parameters have been included. The image below encapsulates the results obtained from the model:

accuracy: 71.16% +/- 6.53% (micro average: 71.15%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	165	70	70.21%
pred. negativeReviews	78	200	71.94%
class recall	67.90%	74.07%	

The project demonstrates the ability of applying text mining in order to classify positive and negative sentiments in the cinema industry (movies) using the K-Nearest Neighbor (KNN) model with an accuracy of 71.16%. Businesses can benefit from the project by analyzing the insights of the audience sentiment. With high specificity (74.07%), the model can accurately detect negative reviews, helping studios identify and address recurring criticisms or audience dissatisfaction.

From a marketing standpoint, analyzing positive reviews helps studios highlight popular elements like performances or storylines in promotions, while negative reviews guide improvements in future productions. Automating review analysis saves time, enabling teams to develop data-driven strategies to enhance audience targeting and engagement.

Appendix:

K-Nearest Neighbors: K = 5

accuracy: 59.49% +/- 7.80% (micro average: 59.45%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	140	105	57.14%
pred. negativeReviews	103	165	61.57%
class recall	57.61%	61.11%	

K-Nearest Neighbors: K = 7

accuracy: 63.97% +/- 6.37% (micro average: 63.94%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	154	96	61.60%
pred. negativeReviews	89	174	66.16%
class recall	63.37%	64.44%	

K-Nearest Neighbors: K = 11

accuracy: 64.57% +/- 7.89% (micro average: 64.52%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	157	96	62.06%
pred. negativeReviews	86	174	66.92%
class recall	64.61%	64.44%	

K-Nearest Neighbors: K = 23 (model chosen)

accuracy: 71.16% +/- 6.53% (micro average: 71.15%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	165	70	70.21%
pred. negativeReviews	78	200	71.94%
class recall	67.90%	74.07%	

Decision Tree: Max Depth = 1 and Confidence = 0.01

accuracy: 64.71% +/- 8.07% (micro average: 64.72%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	145	83	63.60%
pred. negativeReviews	98	187	65.61%
class recall	59.67%	69.26%	

Decision Tree: Max Depth = 13 and Confidence = 0.05

accuracy: 64.71% +/- 7.67% (micro average: 64.72%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	155	93	62.50%
pred. negativeReviews	88	177	66.79%
class recall	63.79%	65.56%	

Random Forest: Num Trees = 100 and Performance Measure = Gini Index

accuracy: 71.32% +/- 4.55% (micro average: 71.35%)

	true positiveReviews	true negativeReviews	class precision
pred. positiveReviews	133	37	78.24%
pred. negativeReviews	110	233	67.93%
class recall	54.73%	86.30%	