Movie Genre Classification using Title and Description

Team 1

Andrea Mechery, Kevin Connolly, Chetan Khatri, Shubham Patil, Balaji Prasad Thulasiram

April 1, 2023

OPIM 5671

Table of Contents

[Executive Summary 2](#_isnxblljxeb9)

[Dataset 2](#_z82olyy5jf2t)

[Final / Best Model 2](#_l0to9gv4w09r)

[Development 4](#_mxzacko5rhki)

[Data Cleaning & Exploratory Data Analysis 4](#_yjbdrcf1m9ih)

[Initial Modeling 6](#_o9ahobhfssd)

[Pivot 12](#_vn2wgzj0p1nt)

[Conclusion 18](#_dehmcurwtf34)

[References 19](#_vi9iuzt4uun6)

[Appendix A: Data Cleaning 20](#_2u667oi7yl7z)

[Appendix B: Final Model Parameters 23](#_bf8t7dko811e)

## Executive Summary

In the digital era of the movie industry and streaming platforms, the task of genre classification through text mining has become increasingly pivotal. By leveraging text mining techniques to accurately categorize films, platforms can elevate user engagement and satisfaction through personalized content recommendations tailored to individual preferences. However, achieving precise genre classification amidst the vast span of film genres poses unique challenges that necessitate innovative approaches.

This project embarks on a comprehensive exploration of movie titles and descriptions provided by IMDb, and aims to accurately predict the associated genre based on text mining principles and practices learned in class. Through a highly exhaustive trial and error based modeling approach, a final model was derived and included several key nodes. Each of the 10+ iterations conducted throughout this process provided the team with additional information that could be used in the following iteration. The final model outputted a misclassification rate of 26%, implying that it achieved a high level of accuracy in classifying movie genres based on textual data. This, thereby, can enhance the efficiency and effectiveness of content recommendation systems and marketing campaigns within the movie industry and streaming platforms.

## Dataset

The project report utilized a dataset sourced from Kaggle1, originating from IMDb (Internet Movie Database), an online repository containing information on films, television programs, home videos, video games, and streaming content. The dataset consisted of both training and testing data, although only the training subset was employed in the project. It comprised 54,214 records, and consisted of 27 genres.

The dataset was structured with four unique variables presented as "ID ::: TITLE ::: GENRE ::: DESCRIPTION", encompassing:

1. **ID**: A numerical identifier assigned to each record.
2. **Title**: The title of the film.
3. **Genre**: The categorized genre of the film.
4. **Description**: A detailed overview or synopsis of the film.

## Final / Best Model

The final model for this project was derived following over nine rounds of iterations, including a consultation with the professor and a subsequent sharp pivot in approach. This final model comprises several key components, including Text Parsing, Text Filter, Text Cluster, Text Topic, and Metadata nodes, all integrated into a Logistic Regression Model with Backward Selection (see Figure 1). The parameters for each of the nodes can be found in Appendix B. Notably, the final model is characterized by three distinct subsets, each with varying configurations. The first subset incorporates Singular Value Decomposition (SVD) components and achieves the lowest misclassification rate of 0.26. In contrast, the second subset excludes SVDs and records a slightly higher misclassification rate of 0.35. Lastly, the third subset further simplifies the model by eliminating both SVDs and Text Topic Raw features, resulting in a higher misclassification rate of .40.

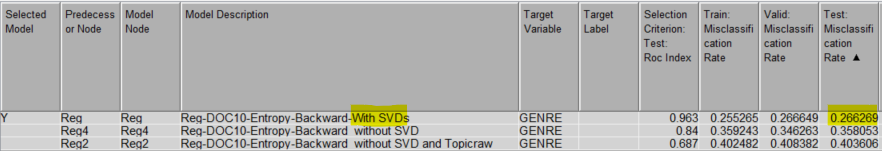


Figure 1. Final Model Results

This progression from complex to simplified models highlights the trade-off between achieving lower misclassification rates and maintaining model interpretability. The first model, incorporating SVD, achieves the lowest misclassification rate of .26, suggesting that capturing latent text structures significantly enhances predictive accuracy, albeit at the cost of interpretability due to the abstract nature of SVD components. The second model excludes SVDs for a simpler approach, yet rises in misclassification to .35, indicating a loss of depth in textual analysis that SVDs afford. The third variant excludes both SVD and Text Topic Raw features, which directly capture thematic or stylistic elements of genres, prioritizing interpretability by relying on the most straightforward features. Yet this model has the highest misclassification of the 3 subsets. Deciding which model to use would depend on the specific needs and constraints for involved stakeholders and the business applications of the model's results.

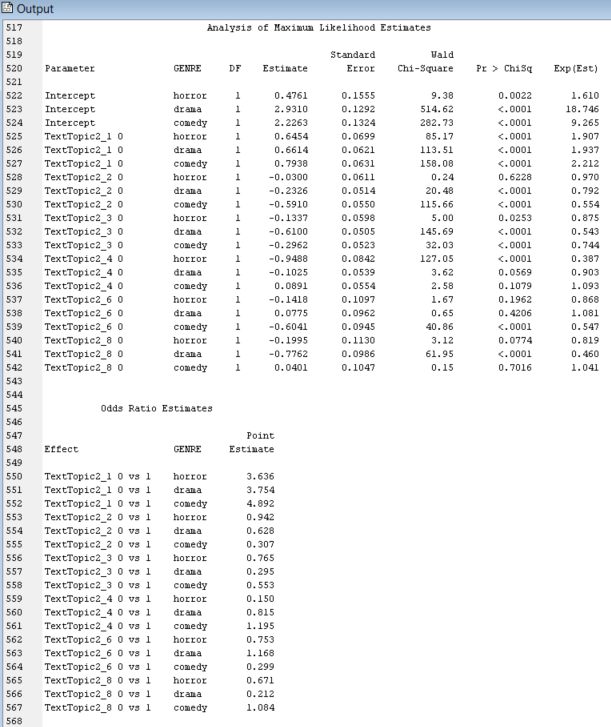


Figure 2. Interpretable Regression (Reg2 - Without SVD & Text Topic Raw) Model Fit Statistics

## Development

### Data Cleaning & Exploratory Data Analysis

Utilizing Python, minimal data cleaning was performed for this project (see Appendix A). This process involved examining the dataset and subsequently identifying and eliminating any duplicate entries, null values, or other irregularities present. Furthermore, the distribution of genres within the dataset showed to include a diverse range in the number of films belonging to each genre. Notably, genres such as Drama, Documentary, and Comedy exhibited substantial representation, each comprising over 7000 records. Conversely, genres like Game-Show, News, and War were among the least prevalent, with fewer than 200 entries each (see Figure 3).

A screenshot of a computer

Description automatically generated

Figure 3. Total list of genres present in the dataset.

The exploratory data analysis phase of the text mining project focused on genre classification, a variety of techniques were employed to gain insights into the dataset. One key approach was the utilization of Zipf's plot, a graphical representation of word frequency against rank order, which revealed an exponentially decaying pattern indicative of the Zipfian distribution. This analysis facilitated the identification of terms with low frequency, typically found at the tail of the plot, which were subsequently removed due to their infrequent occurrence (see Figure 4). Furthermore, a number of documents by frequency chart was generated, illustrating the distribution of documents according to their frequency of occurrence. This visualization showcased a monotonic or linear trend, providing valuable insights into the distribution of documents across different frequencies and aiding in the understanding of the overall structure of the dataset (see Figure 5).

|  |  |
| --- | --- |
| Figure 4. Zipf's Plot | Figure 5. Number of Documents by Frequency Plot |

The attribute by frequency chart offered a comprehensive view of the dataset's attributes sorted by their respective frequencies (see Figure 6). This chart revealed a notable trend wherein a greater number of terms, termed "alpha terms," were observed. Specifically, the chart highlighted that there were 384,421 terms identified as important or frequent enough to retain for further analysis or processing.

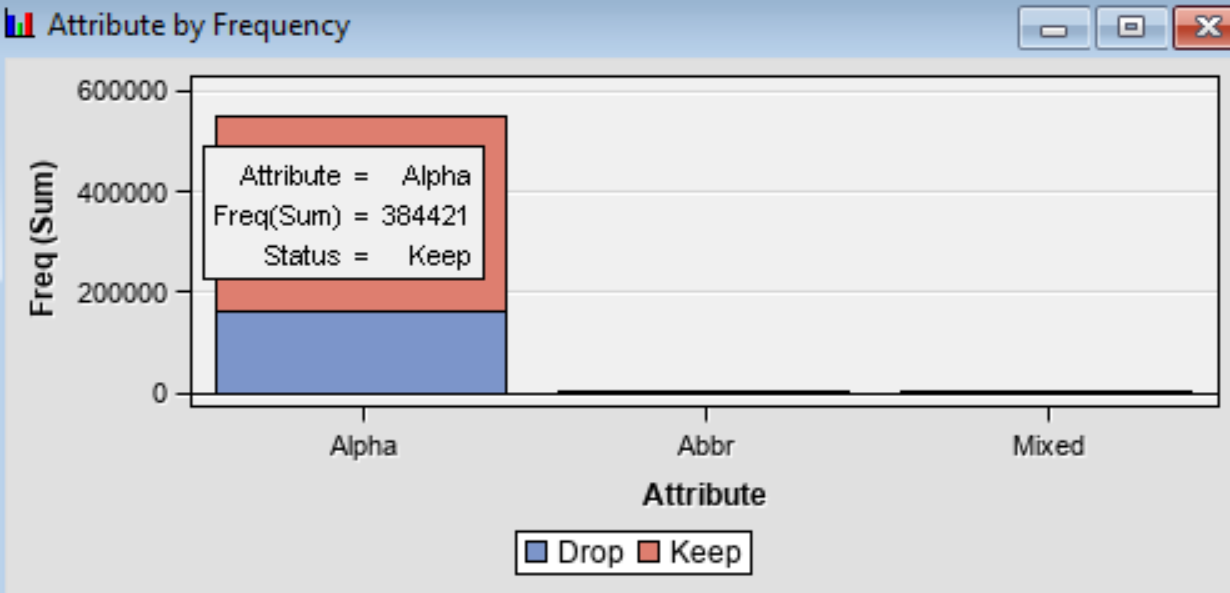


Figure 6. Attribute by Frequency Plot

### Initial Modeling

In the initial phase of the project, the team implemented a base model approach, aiming to assess the performance of various models under default settings. The entire dataset was utilized for this experimentation, partitioned into 80% for training and 20% for validation, with no separate test partition. Text parsing, text filtering, and text clustering were all conducted using default parameters. The models tested included Decision Tree, Regression, Gradient Boosting, Neural Network, and Text Rule Builder, each configured with default settings (see Figure 7). However, it was evident that none of the models yielded satisfactory results, as indicated by the high validation misclassification rates. Consequently, further optimization and exploration were deemed necessary to enhance the model's predictive accuracy and effectiveness.

|  |
| --- |
| Figure 7. (a) Model 1.1 (Base Model) with Default Settings |
|  |
| Figure 7. (b) Model 1.1 Results |

The project team decided to next focus on refining the Text Rule Builder models to enhance model performance following the initial trial. However, concerns regarding low observation counts for certain genres, such as the War genre which had only 132 observations, prompted a cautious approach. Therefore, data partitioning was deferred until the conclusion of model tuning to mitigate potential risks associated with genre imbalance. The progression involved a systematic evaluation of the impact of various Text Rule Build run settings, denoted by parameters such as Generalized Error (VL, L, VH), Purity of Rules (Low, Medium), and Exhaustiveness (VL, L, M, H, VH). This comprehensive analysis allowed for the optimization of the model, prioritizing Training Misclassification as no validation dataset was available at this stage, owing to constraints related to model run time and the aforementioned genre observation limitations (see Figure 8).

| **A screenshot of a chat  Description automatically generated** |
| --- |
| Figure 8. (a) Initial Experimentation of Text Rule Build run settings |
| **A screenshot of a graph  Description automatically generated** |
| Figure 8. (b) Model Results |

The project team then proceeded to explore the Text Rule Builder model, employing various configurations to optimize its performance. The partition ratio was adjusted to 90% for training and 10% for validation to allocate a larger portion of the data for training. Within this framework, modifications were made to the text parsing node, incorporating an amended stop list by adding the top 10 most frequent terms. Additionally, the Text Filter module was manipulated to assess term weights using the default settings, Mutual Information, Entropy, IDF, and None. The Text Cluster aspect was refined by experimenting with different Singular Value Decomposition (SVD) resolutions, comparing Low versus High resolutions with a maximum SVD dimension of 100. Moreover, the Text Rule Builder models were analyzed based on varying levels of Generalization Error (Medium versus Low), Purity of Rule (Medium versus High), and Exhaustiveness (Medium versus Very High). These adjustments aimed to refine the model's predictive capabilities and enhance its performance in genre classification tasks. Despite extensive iterations and adjustments made to the Text Rule Builder models, no iteration yielded satisfactory results, and the misclassification rate remained unacceptably high (see Figure 9).

|  |
| --- |
| Figure 9. (a) Model 1.2: Text Rule Builder with Different Term Weights |
|  |
| Figure 9. (b) Model 1.2 Results |

Throughout the implementation of the prior models, the project had been encountering a significant performance challenge with an extensive 7-hour runtime per model. This prolonged runtime was primarily attributed to the parameter setting of Exhaustiveness being designated as Very High, posing a considerable obstacle to efficient processing. In response to this challenge, the team devised a solution aimed to mitigate the computational burden without compromising the integrity of the dataset. By leveraging a method to proportionately undersample the dataset, the team successfully reduced its size while ensuring the preservation of both majority and minority classes. Specifically, a Sample Node was implemented, resulting in a more manageable dataset for subsequent analyses (see Figure 10). This strategic approach facilitated improved efficiency and expedited the model runtime, addressing the performance constraints encountered during earlier iterations of the project.

|  |  |
| --- | --- |
| Figure 10. (a) Original Data | Figure 10. (b) Sampled Data |

The subsequent phase of the project involved implementing the Text Rule Builder model with undersampled data to address the computational challenges encountered previously. To achieve this, the team opted to utilize only 30% of the original dataset, amounting to 54,213 rows spanning across 27 genres. This undersampling was facilitated through the Sample node, configured with settings specifying a Percentage of 30, a Criterion of Proportional Sampling, and a Minimum strata size of 140. Subsequently, the undersampled data was integrated into the Model 1.2 process, enabling the team to proceed with model development and evaluation under more manageable computational constraints. The partitioning scheme allocated 90% of the data for training purposes and reserved 10% for validation, ensuring a robust assessment of model performance. Despite the substantial reduction in model runtime achieved through undersampling techniques, there was no discernible improvement in results attributable to the undersampling process.

|  |
| --- |
| Figure 11. (a) Model 1.3 (Implementation of Sample Node) |
|  |
| Figure 11. (b) Model 1.3 Results |

In the subsequent iteration, the project team focused on implementing the Text Rule Builder model augmented with a User Topic Table, aiming to further enhance genre classification performance. Utilizing the complete dataset comprising 54,213 rows and encompassing 27 genres, the data was partitioned into training (80%) and validation (20%) sets. Standard text preprocessing techniques, used in prior iterations, were employed to prepare the text data for analysis. The team then explored various configurations of the Text Rule Builder model, assessing parameters such as Generalization Error (Very Low), Purity of Rule (Very Low), and Exhaustiveness (Very High). Additionally, a Text Profile Node was introduced to identify 25 terms for each genre, culminating in the creation of a User topic list (see Figure 12c) for incorporation into the model alongside other variables (see Figure 12a). These iterative refinements aimed to optimize the model's ability to accurately classify genres based on textual features to enhance the model’s predictive capabilities.

However, while the train misclassification notably decreased from previous models (from 0.51 to 0.28), the validation misclassification remained unchanged (see Figure 12b). This discrepancy indicated overfitting on the training data and suggested that the model may not be the most optimal choice. Following this, the team sought guidance from the professor to address this challenge and refine the model further.

|  |
| --- |
| Figure 12. (a) Model 1.4 (Implementation of Text Profile Node) |
|  |
| Figure 12. (b) Model 1.4 Results |
|  |
| Figure 12. (c) Text Profile Node Results |

### Pivot

After consulting with the professor, the project team decided to address the highly unbalanced data issue by narrowing down the genres for model training. Instead of working with the entire dataset with all 27 genres, the focus shifted to creating a subset comprising only four genres. This strategic approach aimed to streamline model training processes and ensure a more balanced distribution of data across genres. The selected genres included Drama (13,612), Comedy (7,447), Horror (2,204), and Action (1,315), resulting in a total of 24,578 observations. By reducing the number of genres, the team aimed to expedite model runtimes while still maintaining a representative sample of data for analysis and prediction tasks.

Using the newly pruned dataset containing the four selected genres—Drama, Comedy, Action, and Horror—the project team restarted the modeling process. The dataset was partitioned into training and validation sets, with an 80% allocation for training and 20% for validation. Text parsing utilized default settings, and text filtering incorporated default weight parameters and specified a minimum number of documents at 10. Text clustering employed a Singular Value Decomposition (SVD) resolution set to High, with a maximum SVD dimension of 100. Additionally, the Text Topic module was configured to generate four multi-term clusters. The modeling phase encompassed various techniques, including Text Rule Builder models configured with settings denoting Generalization Error as Very Low, Purity of Rule as Low, and Exhaustiveness as Very High. The models utilized in this step included Decision Tree, Gradient Boosting, Memory-Based Reasoning (MBR), Neural Network, and Regression, with the Regression model employing a Forward selection approach. The analysis revealed significantly improved results compared to previous iterations with Neural Network and Regression emerging as the top-performing models, with the lowest misclassification rates on the validation dataset (see Figure 13).

|  |
| --- |
| Figure 13. (a) Model 2.1 (Base Models using subsetted data) |
|  |
| Figure 13. (b) Model 2.1 Results |

Following the promising performance of the Regression and Neural Network models, the project team conducted further exploration by experimenting with different term weights to optimize results. For this iteration, the dataset was evenly split into 50% for training and 50% for validation. While maintaining default settings for the text parsing node, the team tested various text filter term weights, including Default, Mutual Information, Entropy, IDF, and None, while setting the minimum number of documents to 10. The text cluster node was configured with a High Singular Value Decomposition (SVD) Resolution and a maximum SVD dimension of 100, alongside an exact number of clusters set to 4. Similarly, the text topic node incorporated 4 multi-term clusters. The selected models, Neural Network and Regression, were configured with default settings and a Forward selection model, respectively. The results led to the determination that Entropy was the most effective term weight to optimize model performance, prompting the decision to utilize it for further refinement in subsequent model tuning processes.

|  |
| --- |
| Figure 14. (a) Model 2.2 (Regression and Neural Network With different Term Weights) |
|  |
| Figure 14. (b) Model 2.2 Results |

The team then proceeded to experiment with varying minimum document count thresholds in the Text Filter module. Following a similar process to the previous iteration, the term weight was consistently set to entropy, while the minimum document count was tested at values of 10, 12, 16, 7, and 4 (See Figure 15a). These modifications were applied to both the Neural Network (Default) and Regression (Selection model = Forward) models to assess their impact on performance. This systematic approach aimed to identify the optimal minimum document count setting that would enhance the models' effectiveness in genre classification tasks.

However, varying the minimum document count did not yield significant differences in results, and the team decided to maintain the document count to the default of 10 for subsequent model building. This determination was based on the observation that adjustments to the minimum document count did not substantially impact the performance of the models.

|  |
| --- |
| Figure 15. (a) Model 2.3 (Regression and Neural Network with different minimum number of document count in Text Filter node) |
|  |
| Figure 15. (b) Model 2.3 Results |

Next, the team expanded the scope to include Regression models with both Backward and Forward Selection techniques, as well as Neural Network (NN) and Regression models with and without Singular Value Decomposition (SVD) variables (see Figures 16a). Notably, adjustments were made to the Text Topic node, which now included a configuration of four multi-term clusters, with a selection of User topics. Additionally, a metadata node was introduced to compare models with and without SVD variables, providing insights into the impact of this factor on model performance. These refinements aim to assess the influence of SVD variables on the predictive accuracy of the models.

Based on the results, the Regression model with backward selection outperformed other models in terms of validation misclassification rates, with no significant deviation in ROC index. However, it was noted that removing Singular Value Decomposition (SVD) dimensions from the models led to a decline in performance. This finding suggests that increasing complexity may potentially yield better results, albeit at the cost of model interpretability. Thus, while adding complexity could enhance predictive accuracy, it may also compromise the ease of interpretation of the models.

|  |
| --- |
| Figure 16. (a) Model 2.4 (Neural Network and Regression models with Both Backward and Forward Selection & with/without SVD variables) |
|  |
| Figure 16. (b) Model 2.4 Results |

The final iteration, focused on comparing Complexity versus Interpretability using the Regression model with Backward Selection. This approach aimed to provide insights into the trade-off between model complexity and interpretability, shedding light on the impact of SVD variables on predictive accuracy and the overall effectiveness of the Regression model in genre classification. Two regression models were executed: one with Singular Value Decomposition (SVD) and one without (see Figure 17a). Notably, for this iteration, the dataset was partitioned into 40% training, 30% validation, and 30% test, allowing for a thorough evaluation of model performance across different phases.

The conclusion drawn from the final iteration indicates that Model 2.5 achieved the most favorable outcomes, exhibiting superior performance with and without Singular Value Decomposition (SVD) dimensions. This model achieved the lowest misclassification rate recorded throughout the project, standing at .266 (see Figure 17b). These results affirm the effectiveness of the Regression model with Backward Selection in optimizing predictive accuracy.

|  |
| --- |
| Figure 17. (a) Model 2.5 (Regression models with Backward Selection & With/Without SVD variables) |
|  |
| Figure 17. (b) Model 2.5 Results |

## Conclusion

This genre classification text mining project highlights an intricate dynamic between model complexity, interpretability, and predictive accuracy within the context of the movie industry and streaming platforms. Through iterative refinement, the project successfully developed a logistic regression model capable of accurately classifying movie genres. The final model, with a misclassification rate as low as 26%, underscores the potential of text mining techniques in optimizing content recommendation systems. Moving forward, continued research and development in this field will be crucial for further enhancing user experiences in digital entertainment.

## 

## References

1. *Genre Classification Dataset IMDb*. (n.d.). Kaggle. Retrieved April 1, 2024, from <https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb>

## 

## Appendix A: Data Cleaning

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 

## Appendix B: Final Model Parameters

The parameters for each of the nodes involved are as follows:

* Text Parsing

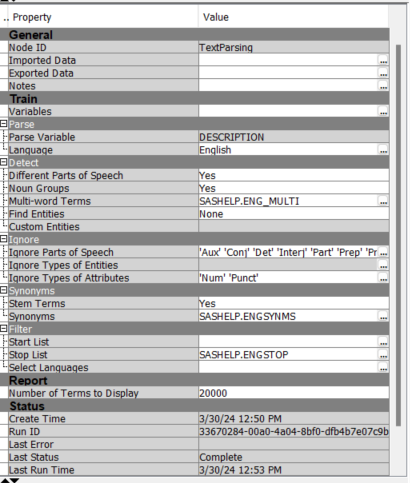


Figure B1. Parameters for the Text Parsing Node

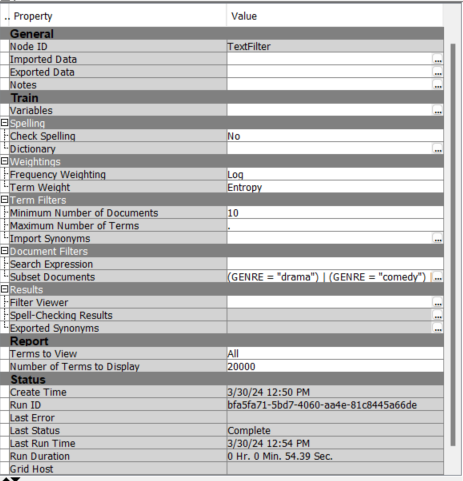
* Text Filter: The term weight of Entropy was decided in Model 2.2

Figure B2. Parameters for the Text Filter Node

* Text Cluster

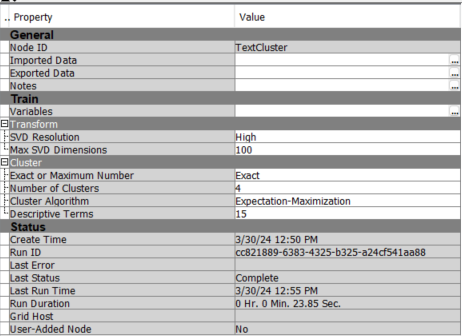


Figure B3. Parameters of Text Cluster Node

* Text Topic

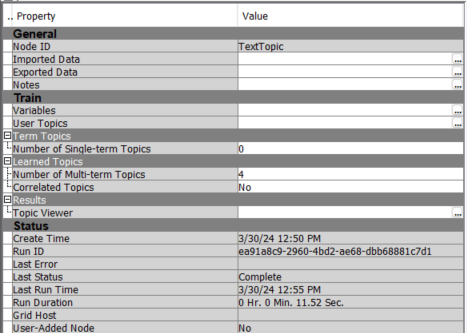


Figure B4. Parameters of Text Topic Node

* Metadata

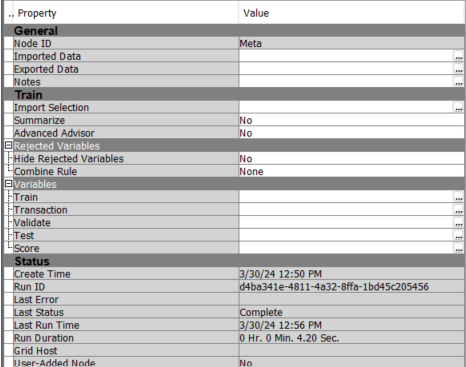


Figure B5. Parameters of Metadata Node

* Regression Model

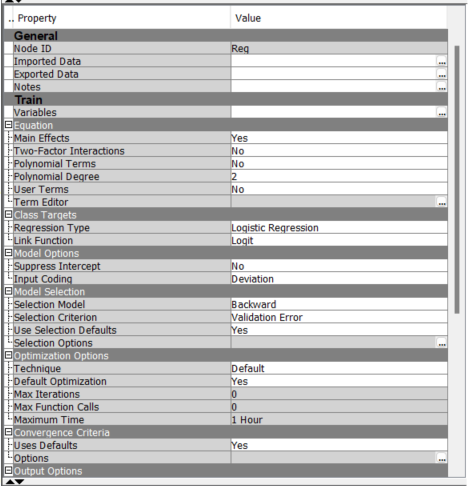


Figure B6. Parameters of Regression Node