**THE TRUTH GUARD:**

**FOR RELIABLE NEWS CLASSIFICATION.**

**FORECASTING  
GLOBAL SUPERSTORE   
TEAM-1**

**Kanchan Sharma**

**Geethasree Madiraju Nagaraju**

**Pothineni LahariPriya**

**Zhexin Zhang**

**Roshita Seelam**

**Table of Contents:**

[**Problem Statement 2**](#_q4ccldj6vhyh)

[**Data description 3**](#_1cibl11h9quy)

[**Data Exploration 5**](#_8vjzmlx7d21x)

[**Data Preparation 8**](#_ijd56lqxr3w)

[**Text Parsing 9**](#_vf030qehcnp1)

[**Text Filter 12**](#_2p9bqkdy34ih)

[**Text Clustering 17**](#_60rn906ywrgb)

[**Topic Modeling 20**](#_rmxvjqyg9yjf)

[**Model Comparison(models) 34**](#_wk0vzx3gbujd)

[**Recommendations 42**](#_6a9fscbdkrii)

### 

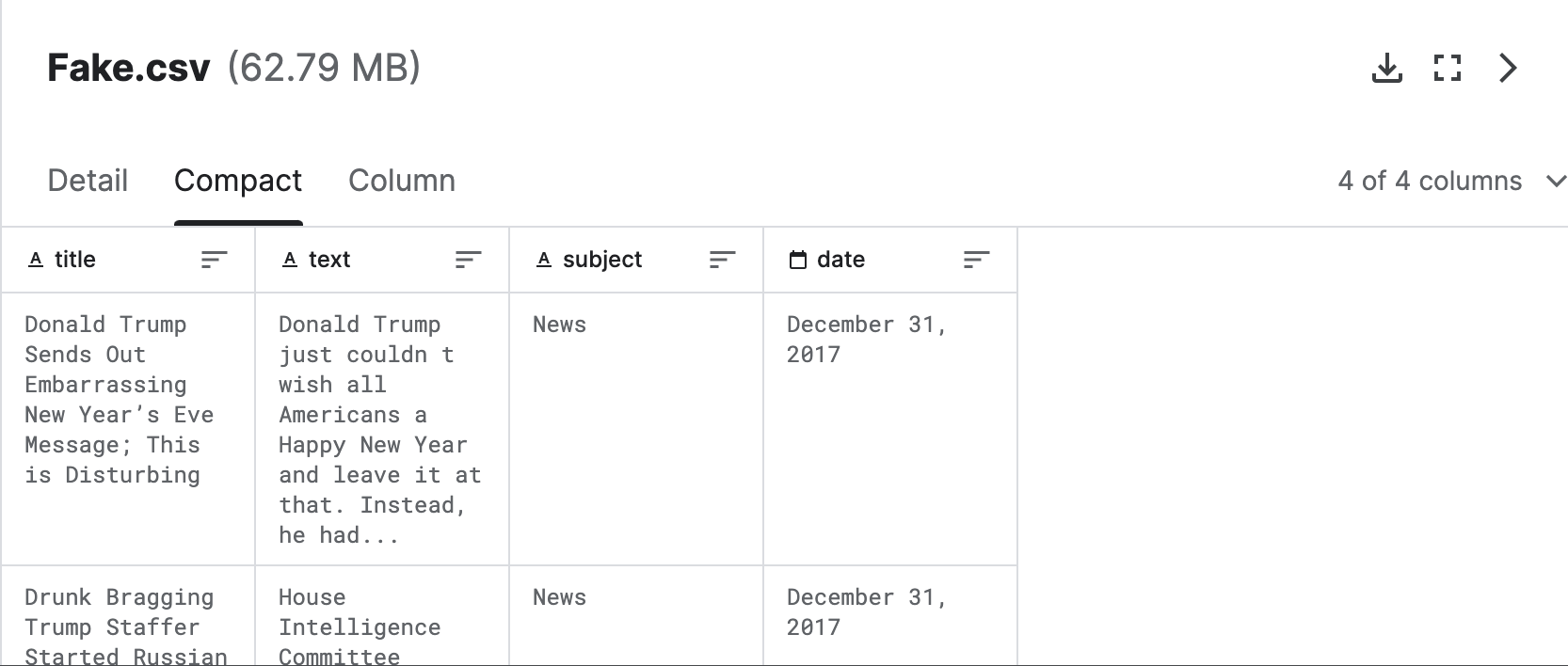
#### Problem Statement

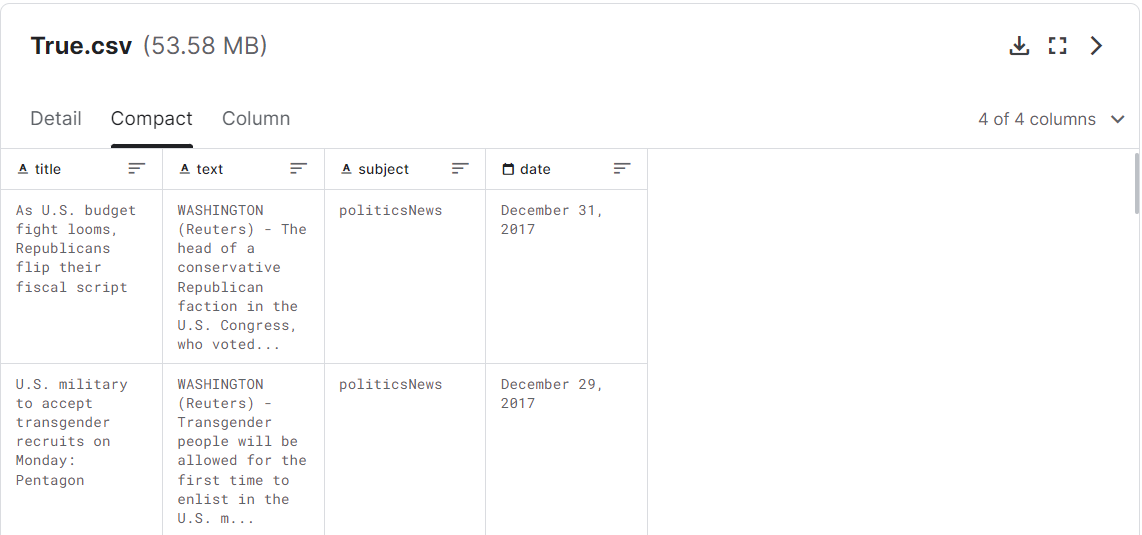
In the past, news was only spread through television and newspapers, and journalists had great control over the accuracy of news. In today's information society, news spreads rapidly through various channels such as social media, news websites, and blogs, resulting in a surge in the spread of misleading news content. Misleading and fake news can mislead public opinion and harm policymaking, social harmony, and citizens’ property. More and more news agencies and social media companies notice this problem and set strict gatekeeping processes to Identify and block fake news. However, Using gatekeepers to identify the authenticity of news is inefficient and likely to be biased. Our group aims to build a model that can classify news articles as "real" or "fake" based on news titles and text descriptions. Our final model and result can help social media companies avoid misinformation and ensure the reliability of news posted on social media.

* Classify news as real or fake. Identify fake news from the dataset
* Identify the key term of fake news

#### Data description

The Dataset is a combination of two files Fake.csv and True.csv, consisting of fake news articles and True news articles respectively in them.





**Title**:

Description: This column contains the title or headline of the news article.

Purpose: The title provides a concise summary of the news and can be important in understanding the main topic or theme of the article. It is often the first thing readers see and can influence their decision to read further.

**Text**:

Description: This column contains the main body of the news article, which includes the detailed content and information.

Purpose: The text column is where the bulk of the information and context is found. It is essential for analyzing the content of the news article and making judgments about its accuracy or authenticity.

**Subject**:

Description: This column indicates the subject or category to which the news article belongs.

Purpose: Categorizing news articles into subjects can help in organizing and classifying the data. It can also be useful for certain types of analyses, such as identifying patterns within specific subjects.

**Date**:

Description: This column contains the date when the news article was published.

Purpose: The publication date is crucial for tracking the timeline of news events. It can also be useful for temporal analysis, trend detection, and understanding the relevance of the news within a specific time frame.

In the context of your fake news detection project, the Title and Text columns will be the primary sources of information for assessing the content and authenticity of the news articles, while the Subject and Date columns can provide additional context for analysis or classification tasks.

Dataset Link

<https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset>

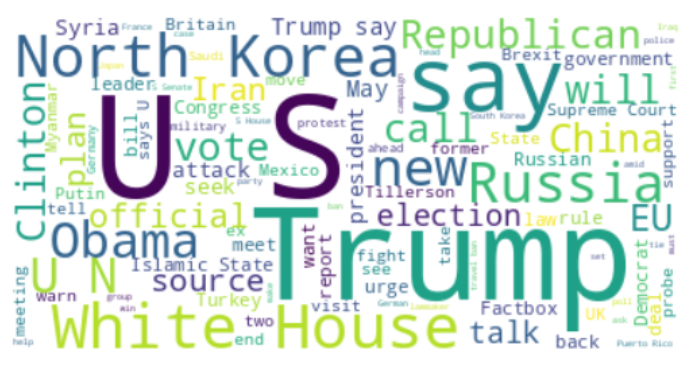
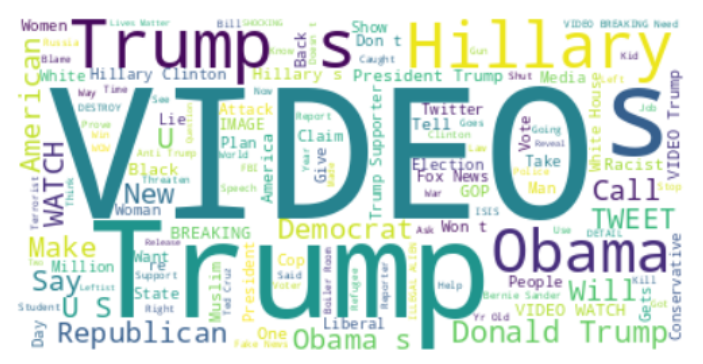
#### Data Exploration

Word Cloud Analysis:

Word cloud analysis is a visual representation technique that illustrates the most frequently occurring words in a given text dataset. The size of each word in the cloud is proportionate to its frequency in the text, enabling the identification of words that occur more frequently and highlighting the linguistic characteristics of the dataset. This method provides an intuitive way to recognize and comprehend the most prominent terms, facilitating the identification of patterns or trends.

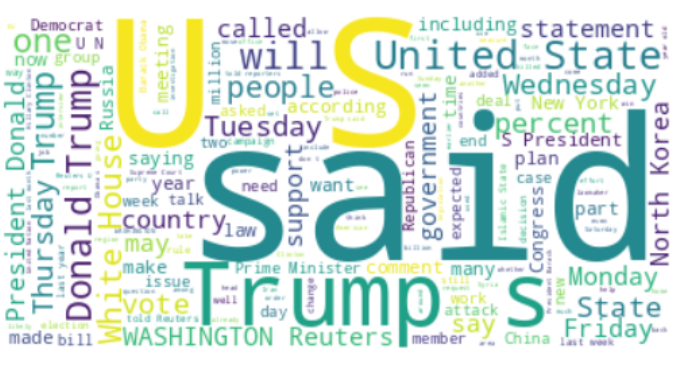
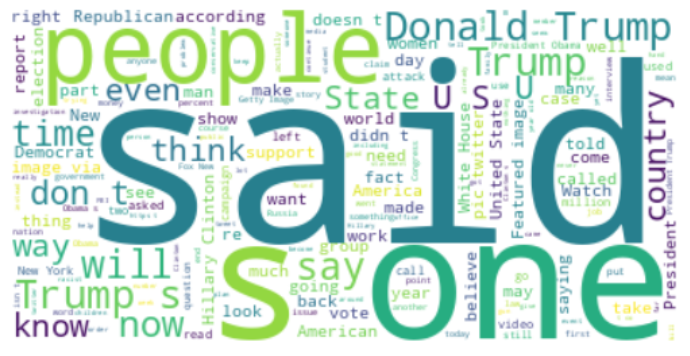
As we possess both titles and text for both fake and real news, we have generated word clouds for both 'title' and 'text' data to visually analyze and compare the language used in fake and real news.

**Word Cloud of Fake News Titles: Word Cloud of Real News Titles:**



Observing the word clouds reveals that fake news titles frequently include terms such as 'video,' which are not commonly found in real news titles.

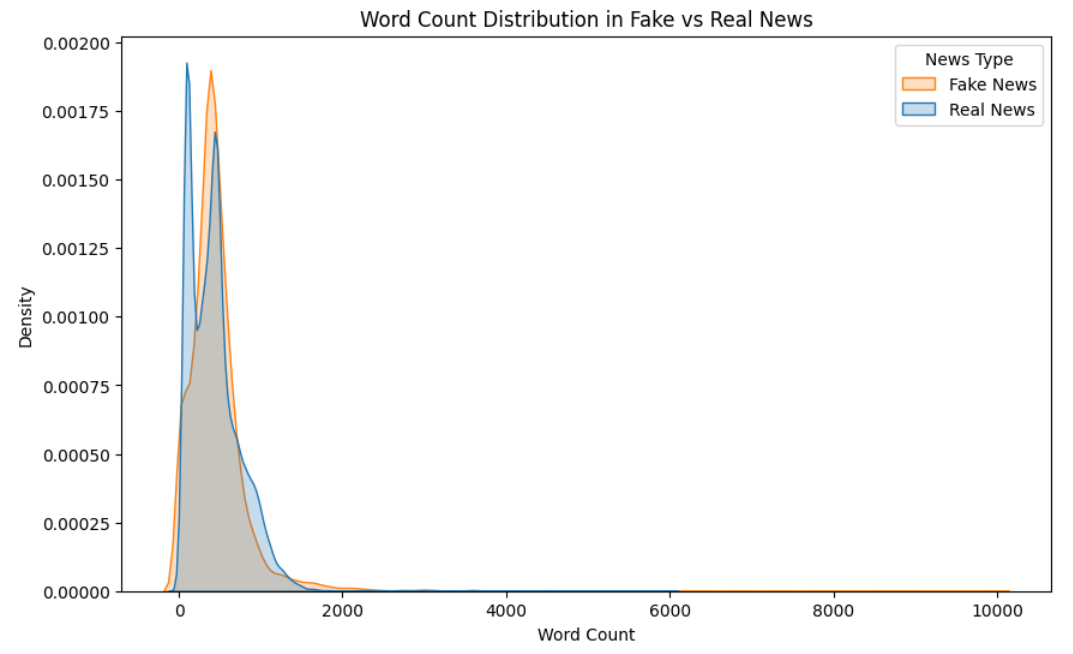
**Word Cloud of Fake News Text: Word Cloud of Real News Text:**

****

Analyzing the word clouds of the text feature reveals that words such as 'people' and 'country' are more commonly found in fake news compared to real news. The word cloud is based on data without text preprocessing, such as stop word removal and lemmatization. Consequently, redundant words like 'say' and 'said' are treated as distinct. However, by cleaning the text before feeding it to the model, we can discern the specific words that occur more frequently in fake news, providing valuable insights.

**Word Count Distribution of Real vs Fake News ‘Text’:**

Following is a plot to visualize the distribution of word count across fake news and real news articles. It provides insights into the typical length of articles in each category and helps identify potential differences between the two categories.



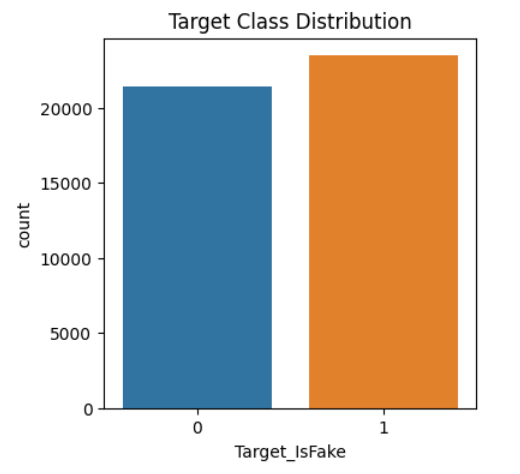
In this context, we can observe that fake news typically tends to be around word count of ~500. Typical News articles text in the dataset tend to be less than 2000 words in length. However there are few articles that are ~10k words in length.

#### Data Preparation

Since our dataset is a combination of two files, Fake.csv and True.csv, containing fake and real news articles, respectively, they need to be merged into a single file that includes both types of articles. A target variable is then created based on whether a news article is fake or real.

Upon merging these datasets and exporting them to a CSV file, few formatting errors were observed. For instance, a sentence like '**They’re Dumb? Science Says “Probably”**' was decoded as '**Theyâ€™re Dumb? Science Says â€œProbablyâ€**'. This was because the csv file format was converting single quotes and double quotes into special characters that are unreadable by SAS Enterprise Miner. Hence importing such data into SAS using File Import Node wasn’t successful, as it couldn't decode these special characters. However, this issue was resolved when the merged file was written in Excel file format."

**Target Class Distribution:**



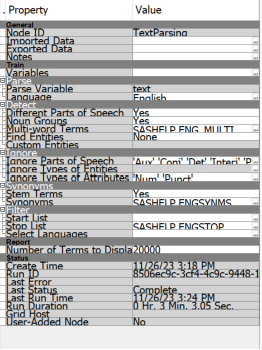
The plot above illustrates the distribution of the target class in our dataset. It is well-balanced, with almost equal representation of both classes. This balanced distribution ensures that our model encounters an equal number of fake and real news articles, contributing to effective classification based on the available data.

After successfully loading the data into SAS Enterprise Miner using File Import node, we have chosen to use a data partition node to split 40% of the dataset into Training, 30% to Validation, rest 30% to Test sets. Furthermore, since our data has two text fields, ‘title’ and ‘text’ for each news article, we chose to experiment with ‘text’ of the news article first, as it has higher word count when compared to ‘title’ and hence has higher likelihood of contributing to the classification problem. In addition to this, we also experimented with the ‘title’ feature alone to have a comparison with the model that used the ‘text’ feature alone. This Model is explained in more detail under the Model Insights section.

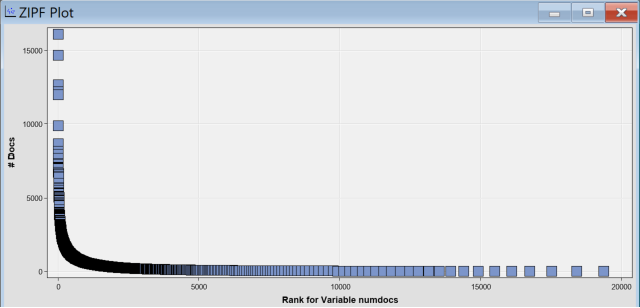
#### Text Parsing

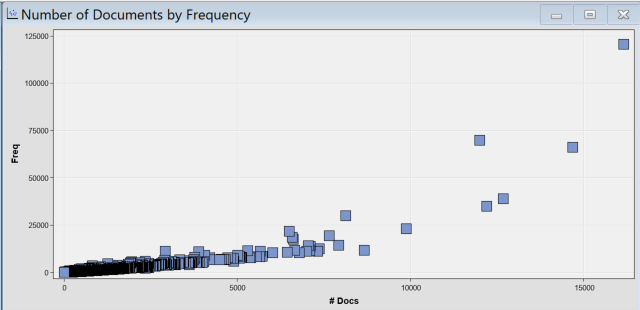
Text parsing is a fundamental step in our text mining project, serving as the initial stage where raw text data is transformed into a structured format for subsequent analysis. For the Text Parsing node, we retained all the default parameters and used the default english stop list available in SAS. We haven’t experimented with custom stop word list, as we observed pretty good model performance with the default stop list.

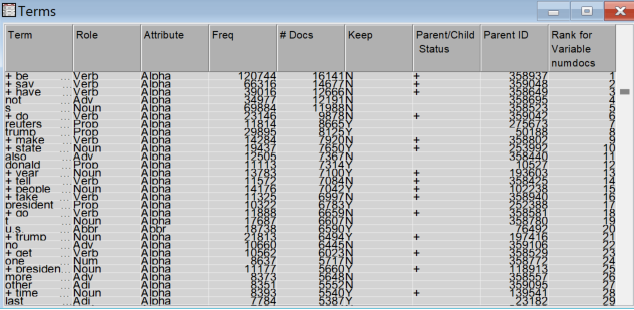
**Properties panel of Text Parsing Node:**



**ZIPF Plot:**

****

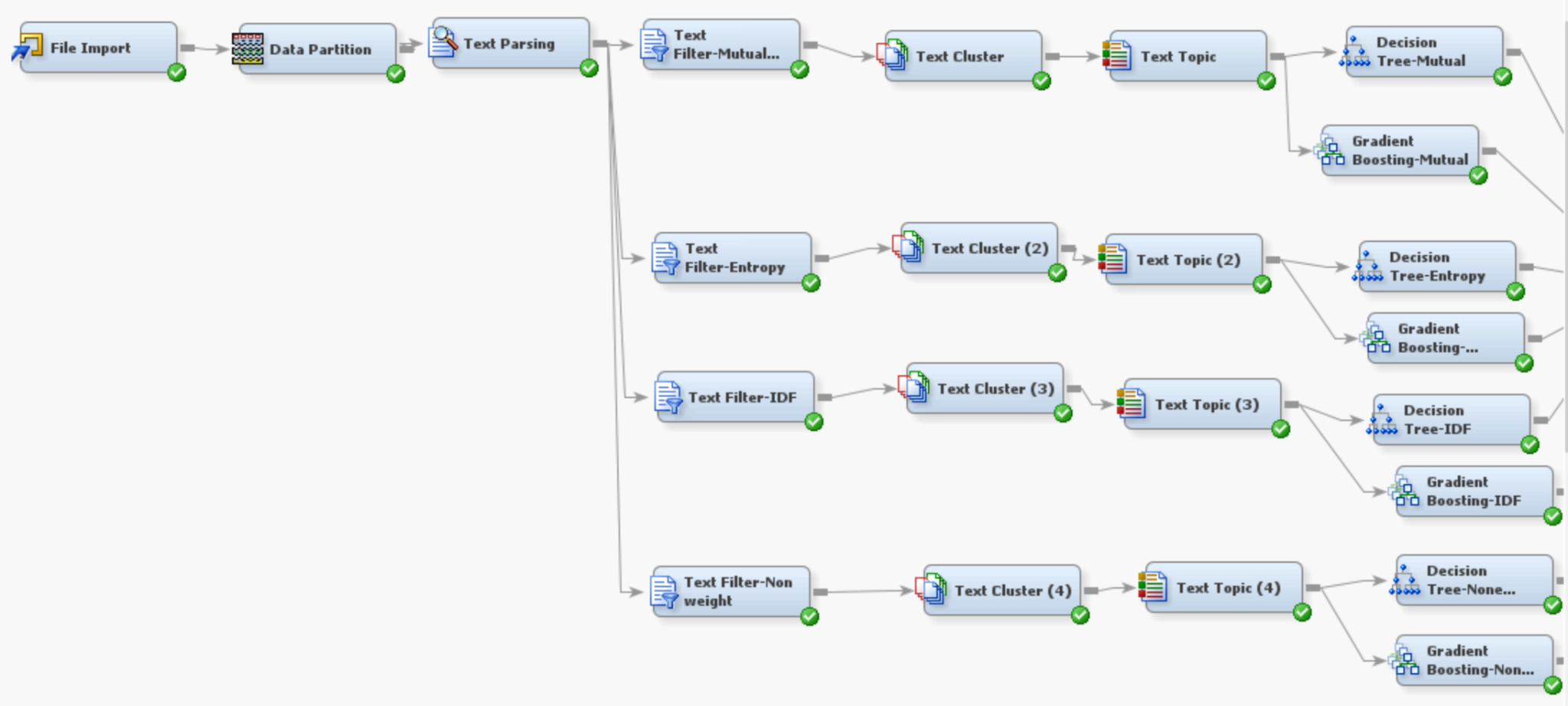
****

****

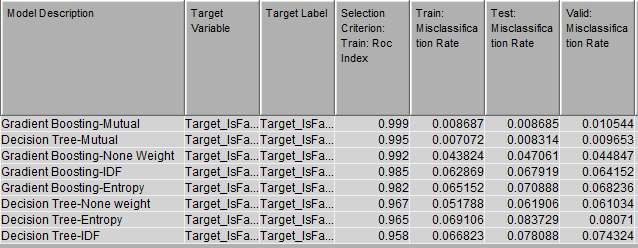
#### Text Filter

Our Group did some experiments on the text filter. We want to try different term weights and frequency weights in order to get the most suitable parameters and increase the performance of the model.

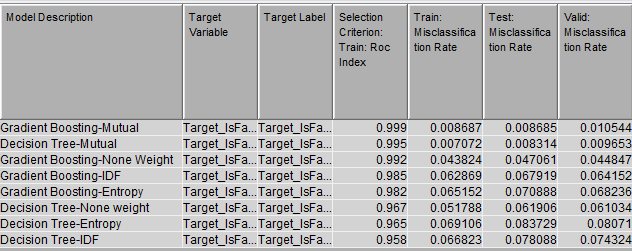
We test four different term weights: Mutual information, Entropy, Inverse document frequency, and non-weight. We also test the performance of three different frequency weighting methods: Binary, log, and default.



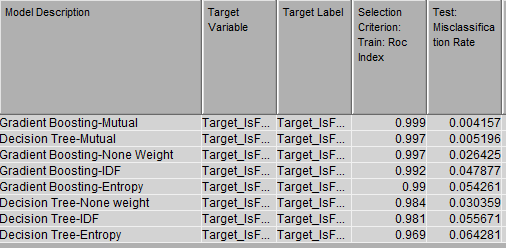
* The result of Log:



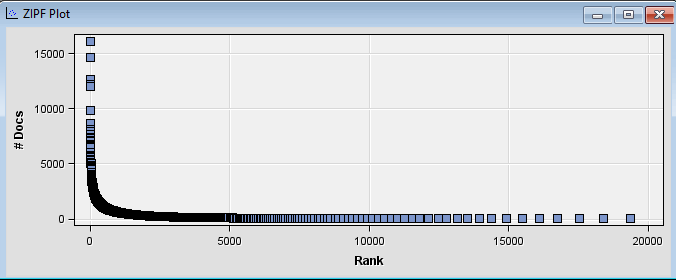
* Default:



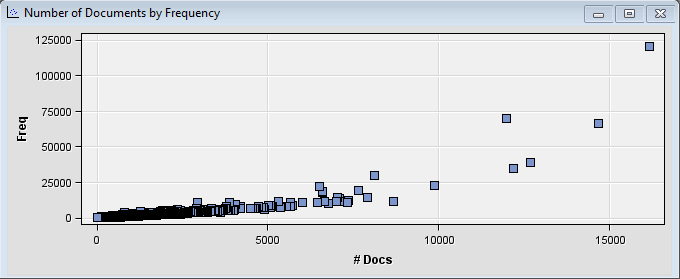
* **The result of Binary:**



* Based on the table above, We found that Binary and Mutual information is better than other options. The models that use binary and mutual information have higher ROC and lower misclassification rates than models that use other text filters.



This ZIPF plot shows the frequency distribution of words. The frequency of a word in the graph is inversely proportional to its rank. A steep initial slope indicates that several words occur very frequently. Those most frequent words could be common English words. In our ZIPF diagram, the top-ranked term is “Be,” the second-ranked term is “Say,” and the third-ranked term is “Have.” Those three words occur more than 13,000 times. However, those three words are meaningless for detecting whether the news is real or fake. We will drop those meaningless terms. The graph also shows a high concentration of terms with low ranks (0-5000). The long tail in the graph suggests that some words appear infrequently. The most infrequent term in this graph is ‘max.’ We may need to consider whether those infrequent (unique) words have an effect on the result.

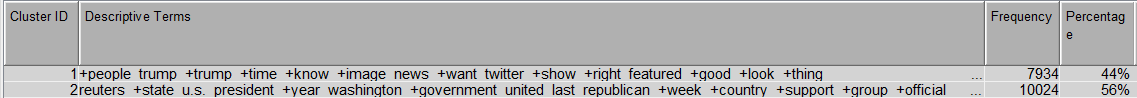


* On the left side of the graph, There is a large number of terms that appear in relatively few documents(0-5000). There is a small number of terms that appear in a large number of documents(10000-15000). There are some outliers, which are terms that appear in a huge number of documents compared to the rest. Those terms are the most common English words and are not meaningful for our classification.

#### Text Clustering

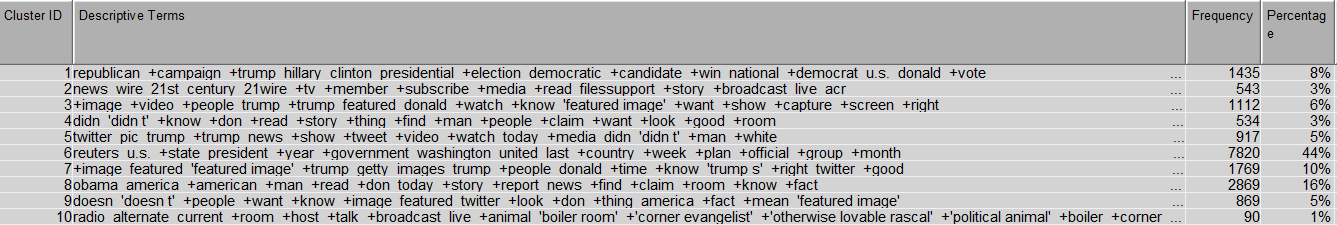
Based on various experiments on Text Filter node, the combination of parameters that led to best performance in terms of ROC\_AUC and misclassification rate are when frequency weight = Binary and Term weight = Mutual information. We also did some experiments on text clustering. We try different combinations of SVD and numbers of clusters. We tried SVD= 30 and 20, and the number of clusters= 2 and 10.

**Number of clusters = 2, SVD= 30, Resolution= High:**

****

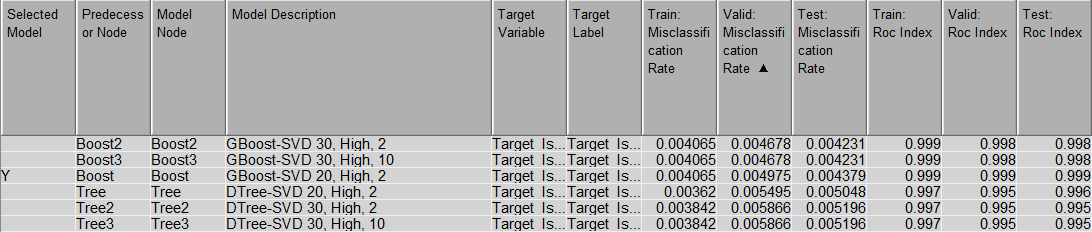
In this experiment, SAS gave us two clusters. The Cluster 1 has 12 descriptive terms. The terms associated with it include “People,” “Trump,” “time,” “know,” “image,” “Twitter,” “show,” “right,” “featured,” “good,” “look,” and “thing.” This cluster 1 represents 44% of the documents in the dataset, indicating that nearly half of the documents are characterized by these terms. The terms in cluster 1 are associated with social media (Twitter) and public figures (Trump and other government officers). Cluster 2 has 15 descriptive terms. Cluster 2 represents 56% of the documents in the dataset. The terms in cluster 2 are associated with news agencies (Reuters), the president, the government, and political parties.

**Number of clusters = 10, SVD = 30, Resolution = High:**

****

In this experiment, SAS gave us 10 clusters. Compared with 2 clusters, 10 clusters gave us more terms to interpret. Cluster 1 makes up 8% of the documents in the dataset, and this cluster is associated with the presidential campaign. Cluster 2 makes up 3% of the documents in the dataset, and this cluster is associated with social media and broadcast. Cluster 3 represents 6% of the dataset and this cluster is related to Trump and multimedia. Cluster 4 accounts for 3% of the documents and it is related to personal opinion. Cluster 5 accounts for 5%, and it is related to Twitter, gender, and race. Cluster 6 represents 44% of documents in the dataset, and it is related to social media (Twitter) and public figures. Cluster 7 accounts for 10%, and it is related to image and Trump. Cluster 8 accounts for 16%, and it is related to Obama and the government. Cluster 9 accounts for 5%, and it is related to images and Twitter. Cluster 1o accounts for 1% and it contains diverse terms.

**Model Results Comparison by changing SVD, number of clusters:**



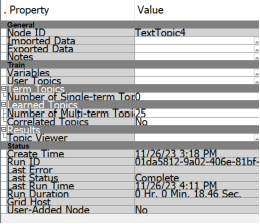
From the above table, we can find that the Boosting model with SVD 30 and 2 clusters has the same ROC and misclassification as the Boosting model with SVD 30 and 10 clusters.

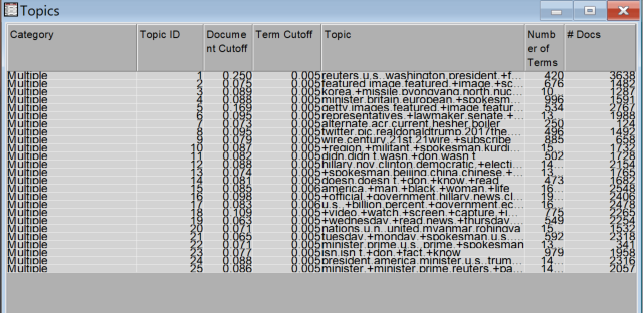
We also found that lower SVD can improve the performance of the decision tree model. On the contrary, High SVD can improve the performance of the Gradient-boosting model.

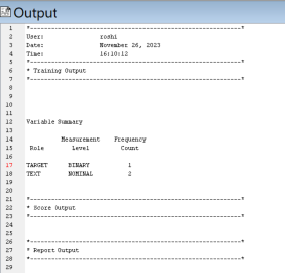
#### Topic Modeling

Our experimentation involved the use of four Text Topic nodes, each configured with default multi-term topics. This decision was based on the desire to achieve a balance between granularity in topic identification and the complexity of the model.

**Parameters:**



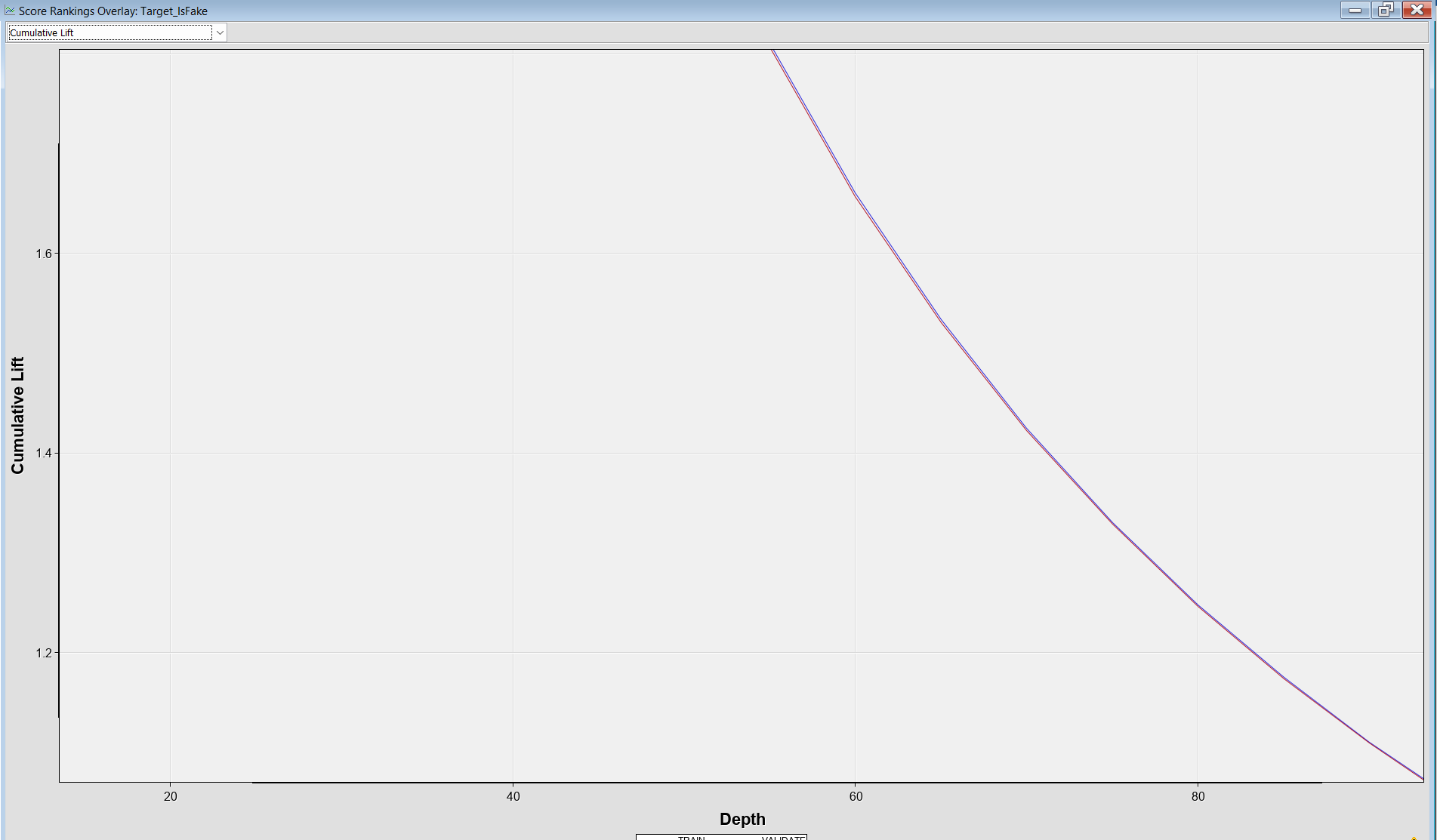
**Result of the log:**



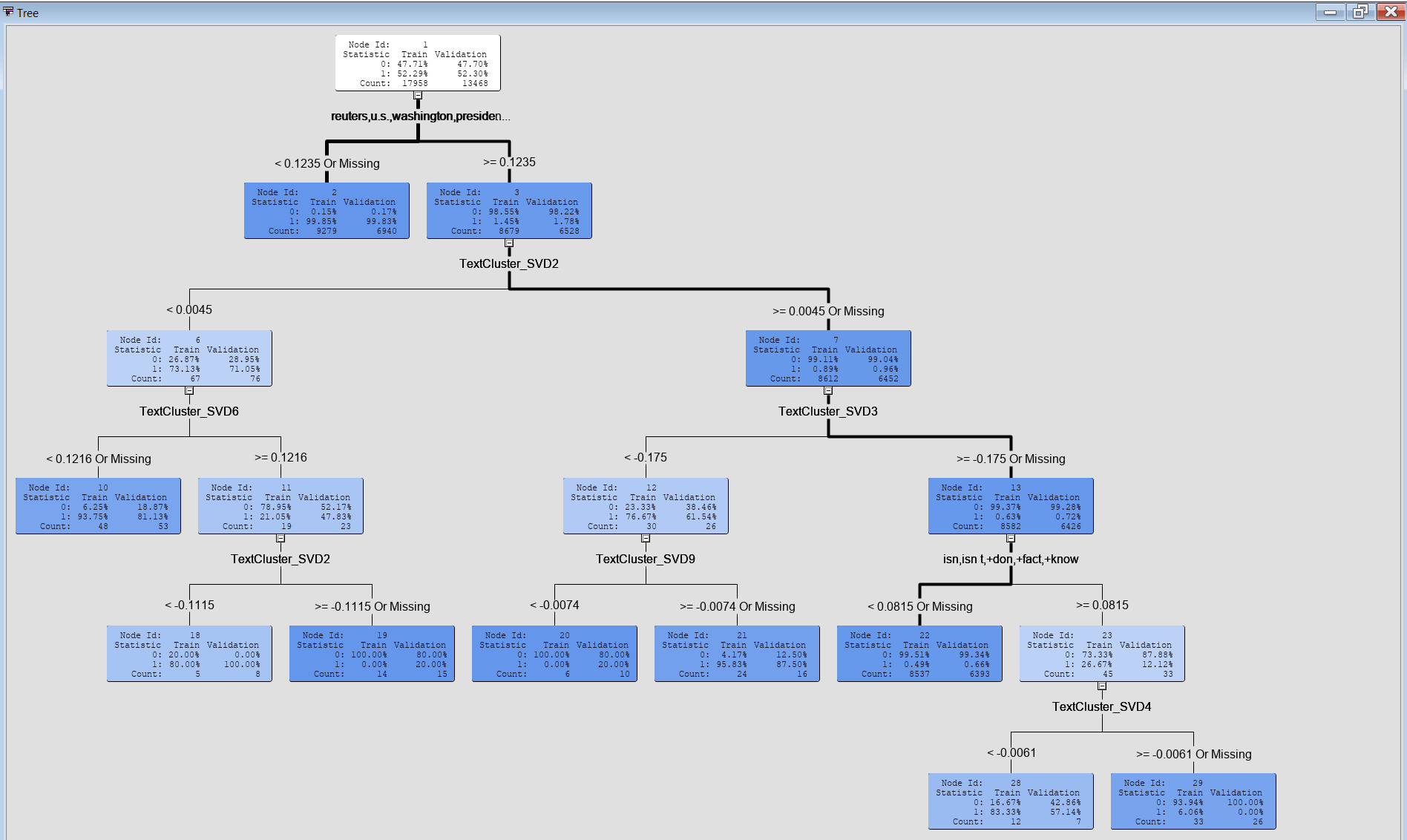
**Model Development:**

**Decision Tree:**

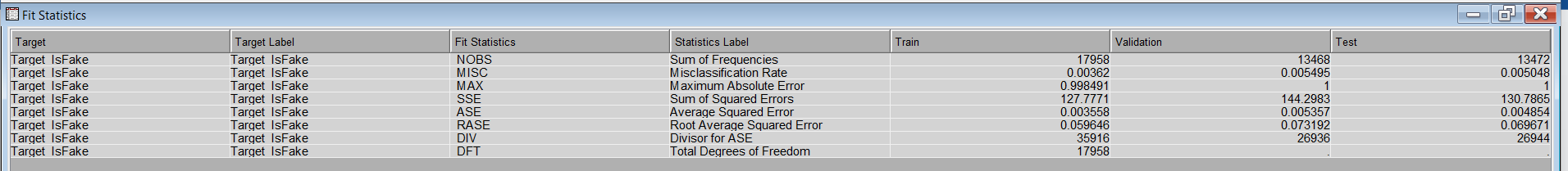
**Cumulative lift:**

****

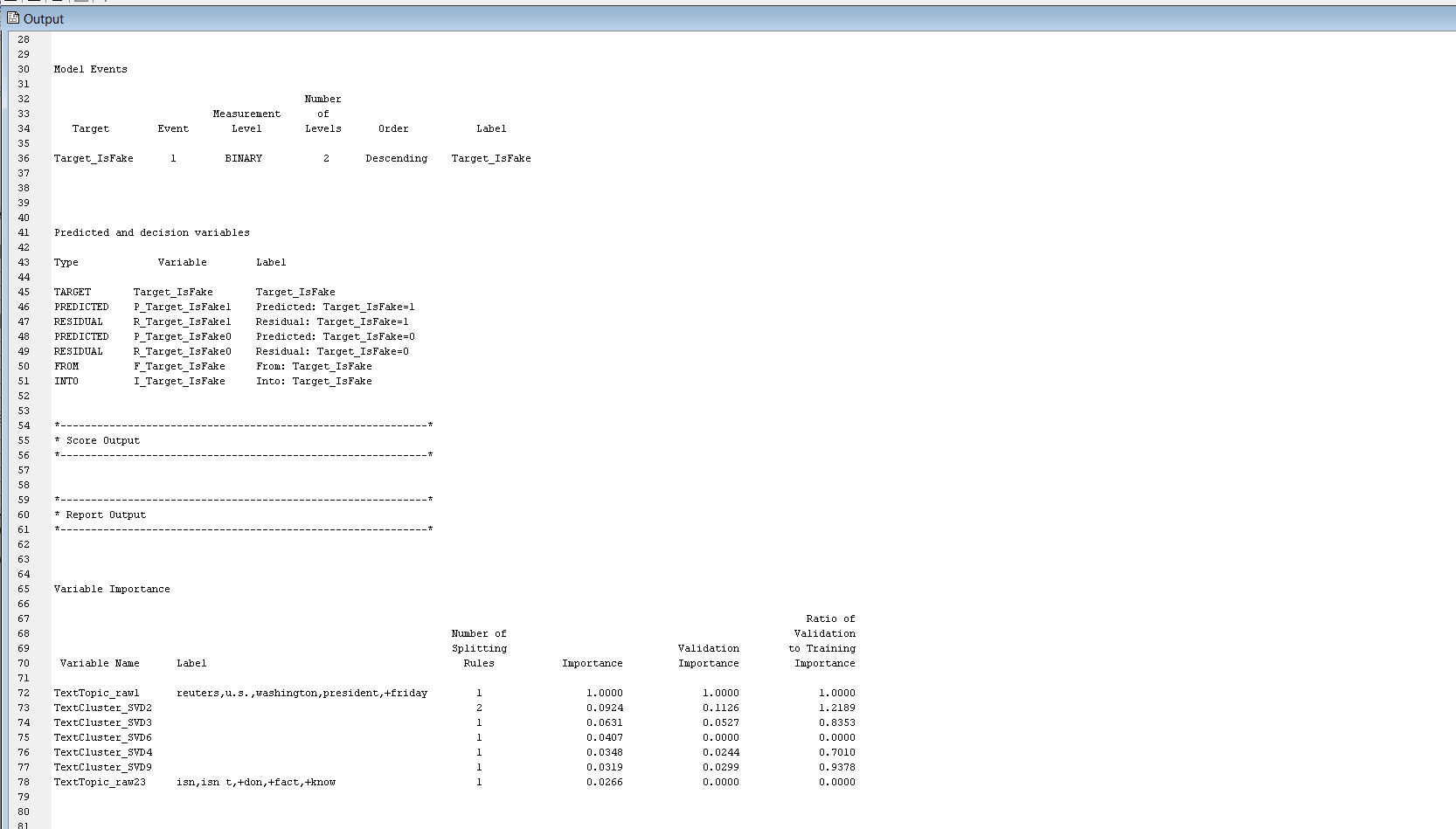
**Tree:**

****

**Fit Statistics:**

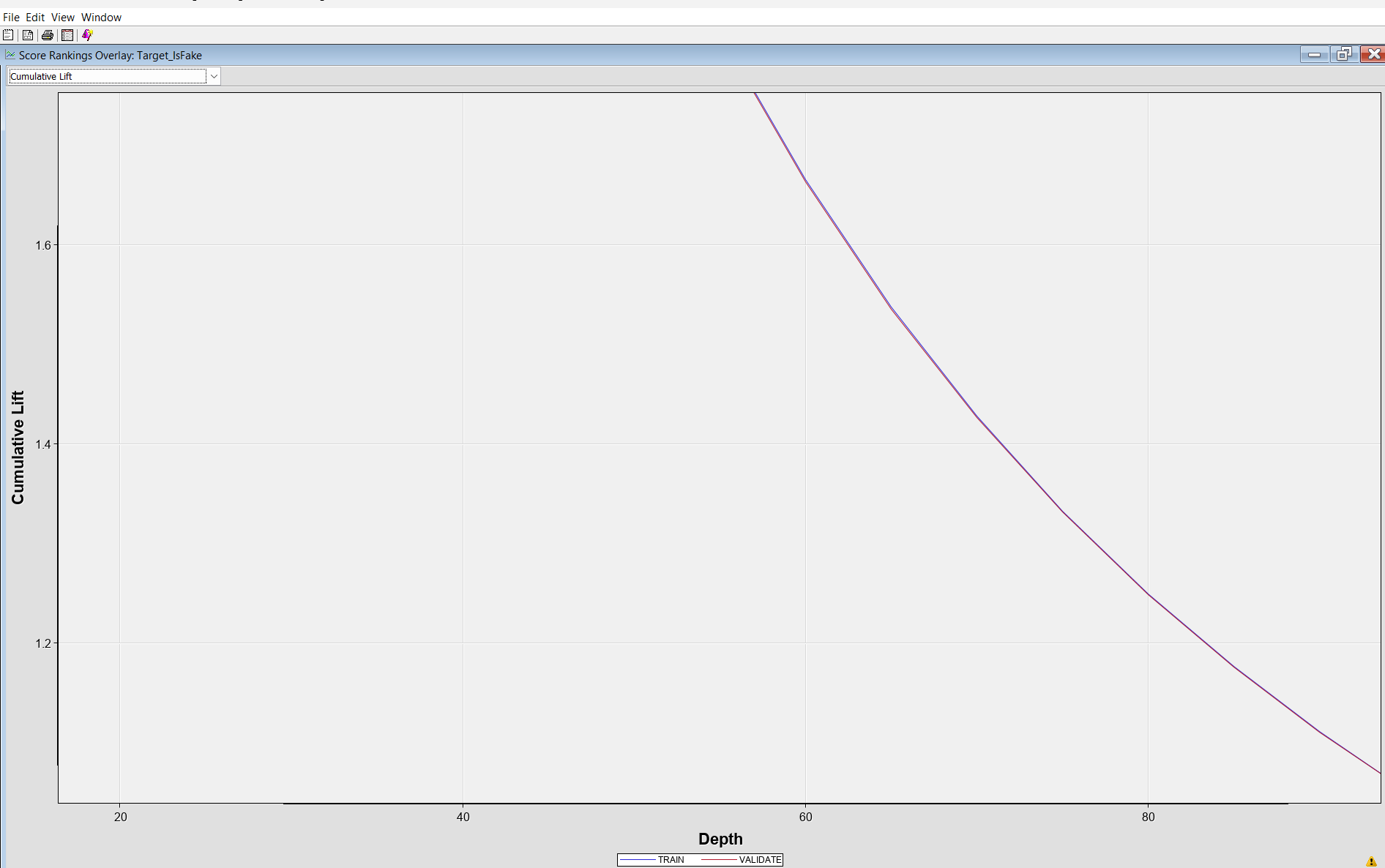
****

**Output of decision tree:**

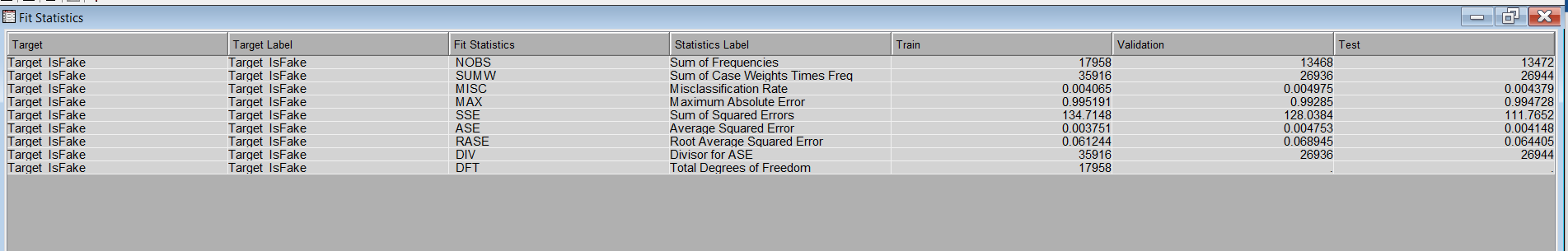
****

**Gradient boosting Model:**

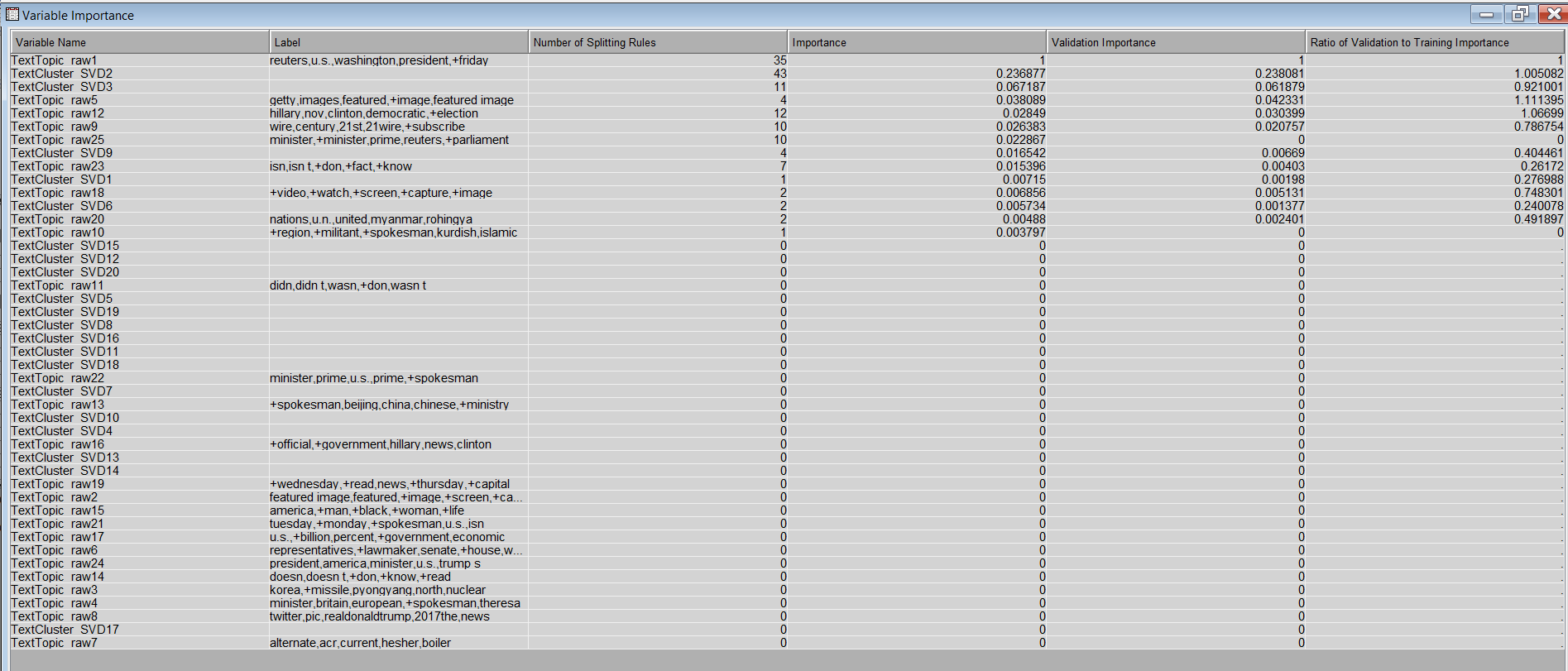
**Cumulative fit:**

****

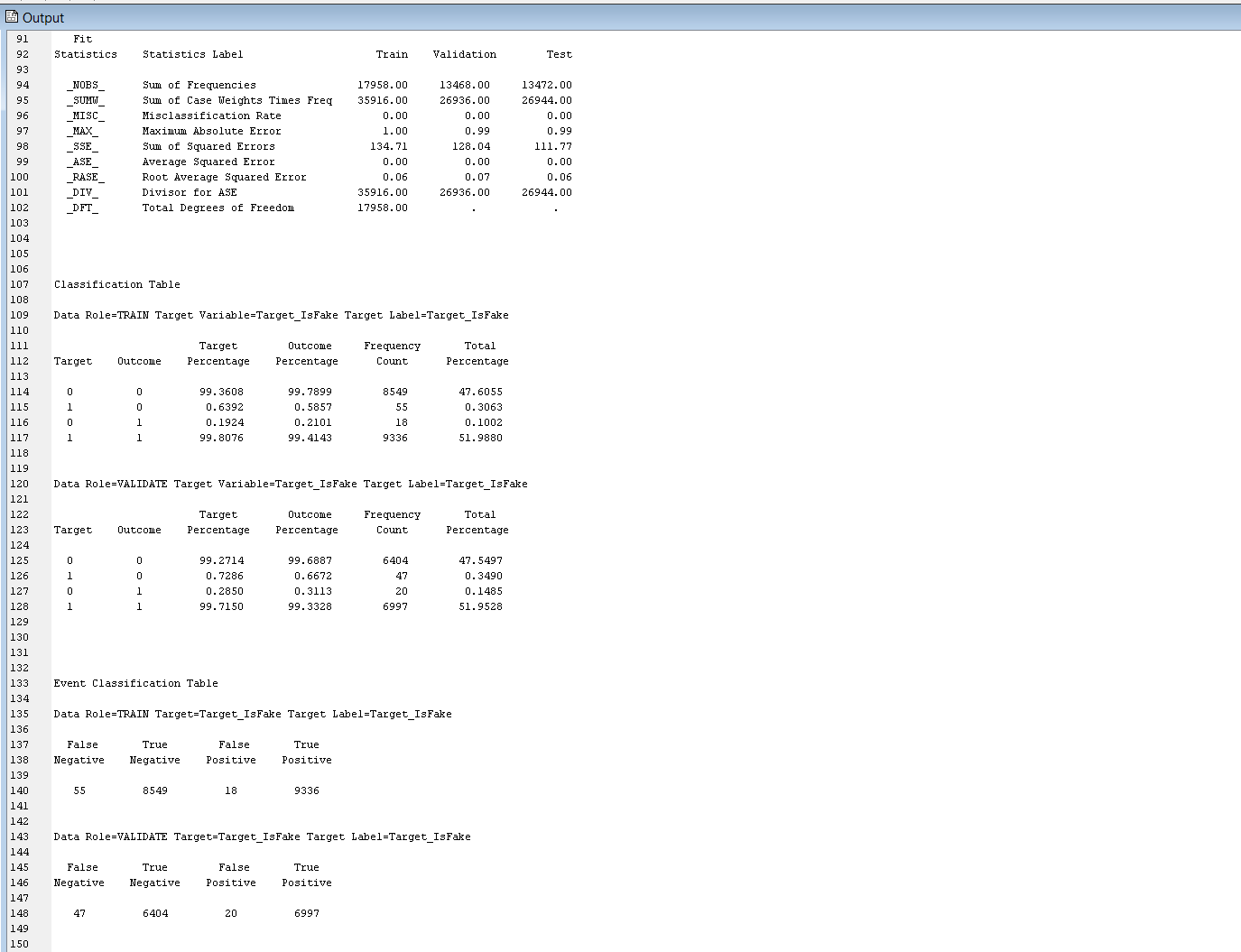
**Fit Statistics:**

****

**Variable Importance:**

****

**Output:**

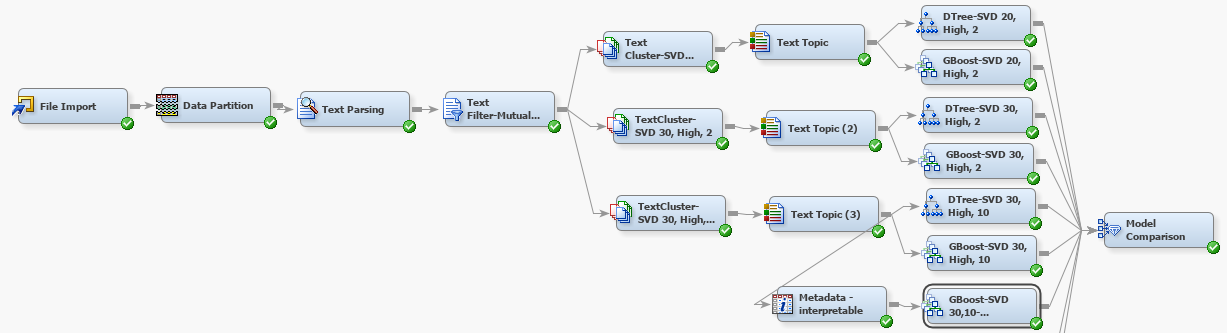
****

**Model Interpretation & Insights:**

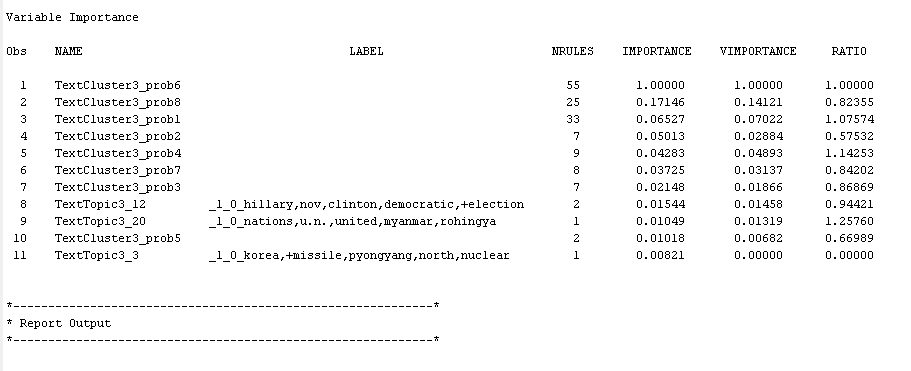
Although we have achieved good performance measures with the chosen parameters and modeling techniques, it's highly difficult to interpret the model and understand what kind of terms are helping the model to distinguish between fake vs real news because of its input SVD features. Hence, we have chosen to build an interpretable model that can help the business understand the key terms that are most useful in detecting fake news and these insights can help any business take necessary actions to prevent the spread of fake news in advance.

While building the interpretable model, we rejected all the SVDs and Topic raw from the input to the model using MetaData Node as they are not interpretable and instead used cluster probabilities and topic weights as input to the Regression node, as it can give insights on words that are contributing most to the classification problem. By doing so, we might lose some predictive power, and based on performance comparison with the base model whose interpretation is not straightforward , we can choose the best model.

With the best performing combination of parameters obtained so far, we added a metadata note to reject all the SVD components and include cluster probabilities and fit the best performing Gradient Boosting model with these input features.

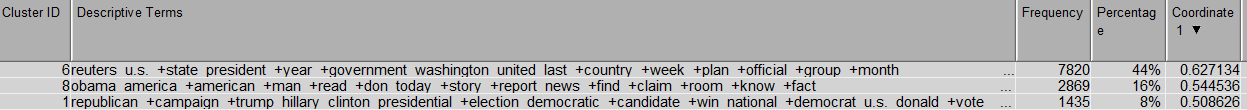
****

Following is the variable importance table obtained from interpretable Model results



From the columns, NRULES, IMPORTANCE, VIMPORTANCE, it can be observed that cluster probabilities 6, 8, 1 contribute to the highest number of decision rules, information criterion etc. Following are the words present in these clusters.

Cluster 6, 8, 1:

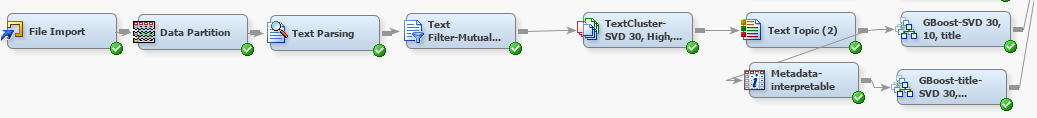


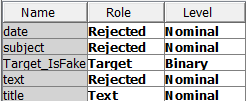
It can be observed that words like **Reuters**, **president**, **government** etc are frequently used words in Fake news articles. Reuters is a news agency owned by Thomson Reuters Corporation. Upon examining a few fake news articles, we found that they have made references to articles or photos published by Reuters to gain more credibility.

Furthermore, Topic 12 had highest importance when compared to all other topics. It can be observed that words like **hillary**, **nov**, **clinton**, **democratic**, **election** signify the most frequent topics around which fake news are created.

**Model with ‘title’ feature alone:**

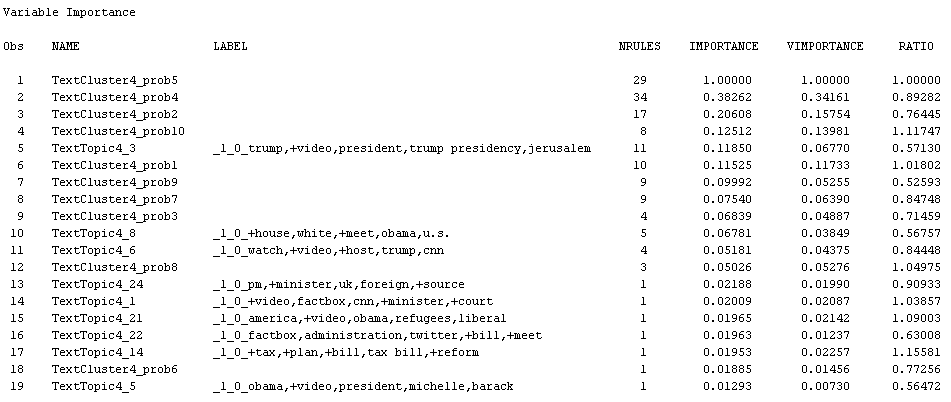
Since our data has two text fields named ‘title’ and ‘text’, we also experimented with the ‘title’ feature alone to see if it can help distinguish between fake and real news. Although the title feature has fewer number of words when compared to text features, understanding what words in ‘title’ feature contribute to classifying a word article as fake can help in driving better business insights and recommendations.



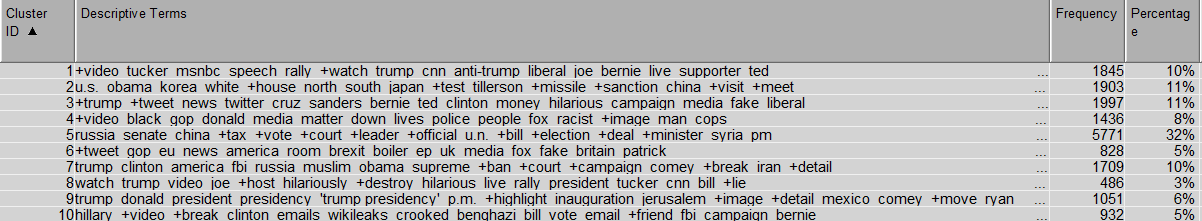


For creating a model with ‘title’ feature alone, use a separate File Import Node, as if there are multiple text data, it considers the column with the longest length. By setting ‘text’ column as rejected in the variables for this node, built two models Gradient Boosting with all features, Gradient Boosting interpretable model excluding **SVD**  features. This interpretable model can help in identifying key words in the ‘title’ column for fake news articles.

Following are the variable importances observed from interpretable model utilizing ‘title’ feature alone



cluster probabilities 5, 4, 2 contribute to the highest number of decision rules, information criterion etc. Following are the words present in these clusters.

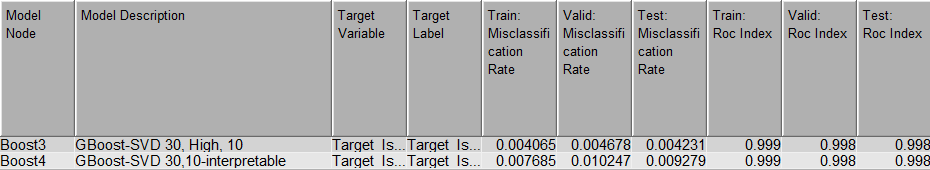


Words like **election, video, bill, senate** etc, are more common among clusters 5, 4, 2.

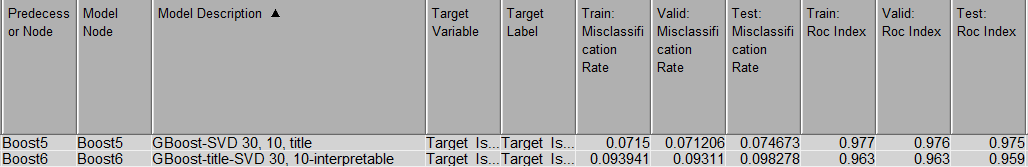
Topic 3 that contains words like **video, president, trump, presidency**, signify the topics around which most of the fake news exist in our datasets.

Following is a comparison of different interpretable models with their base models.

**Using ‘text’ feature alone:**



**Using ‘title’ feature alone:**

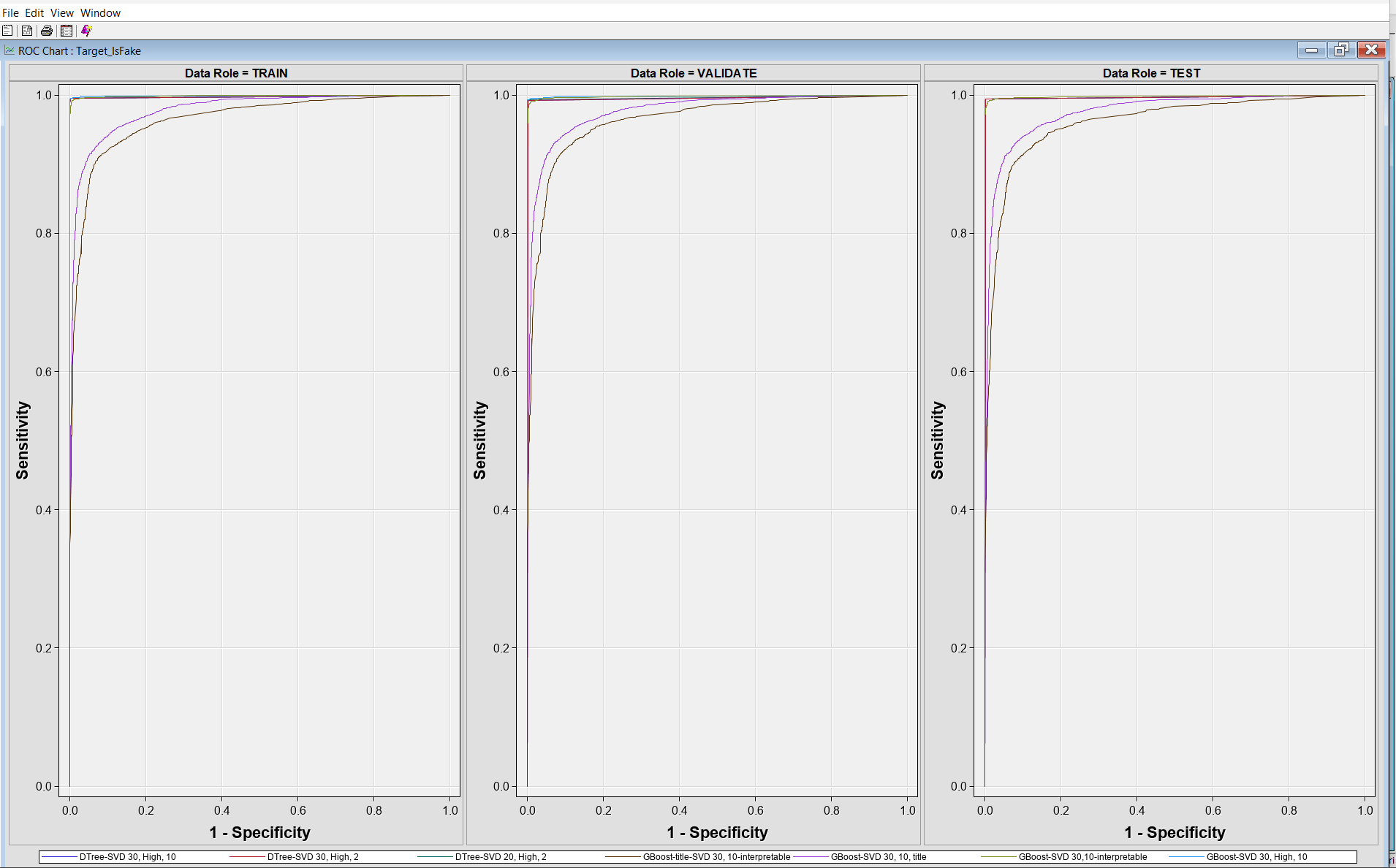
****

For the interpretable models in both the cases (using ‘text’ alone and ‘title’ alone), the misclassification rate is higher, which is expected, as we have rejected SVDs from the inputs to these models. However, these models are interpretable and can be utilized for insightful recommendations.

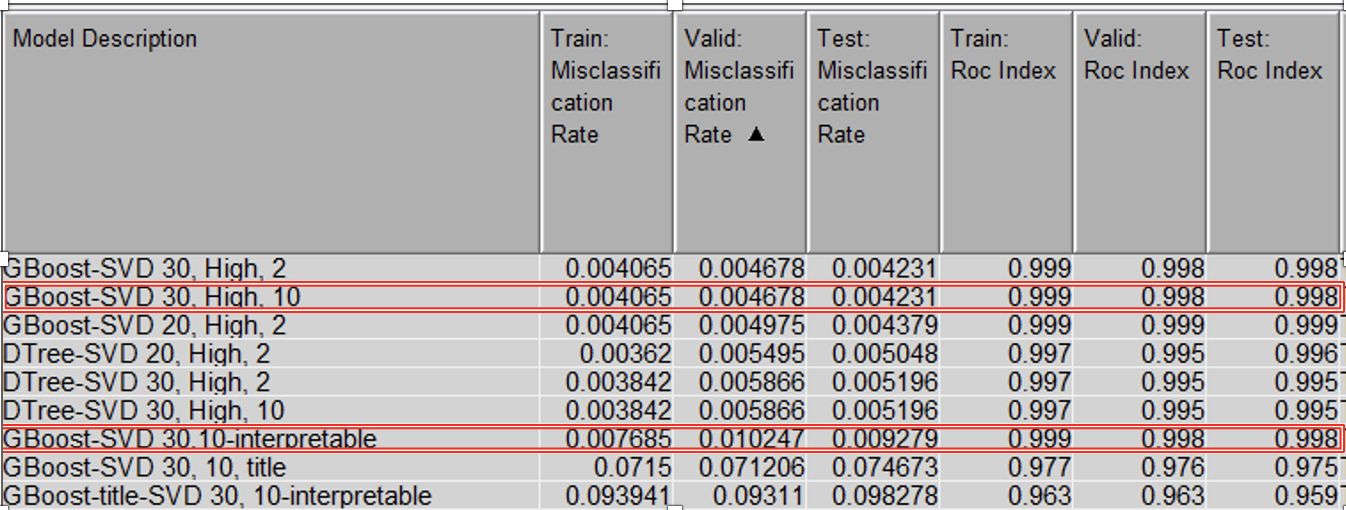
While the words interpreted from clusters and topics could serve as indicators, they cannot be solely relied upon to decide the credibility of news articles. Hence we suggest some actions utilizing these insights in the Business Recommendations section.

#### Model Comparison(models)

**ROC Curve:**

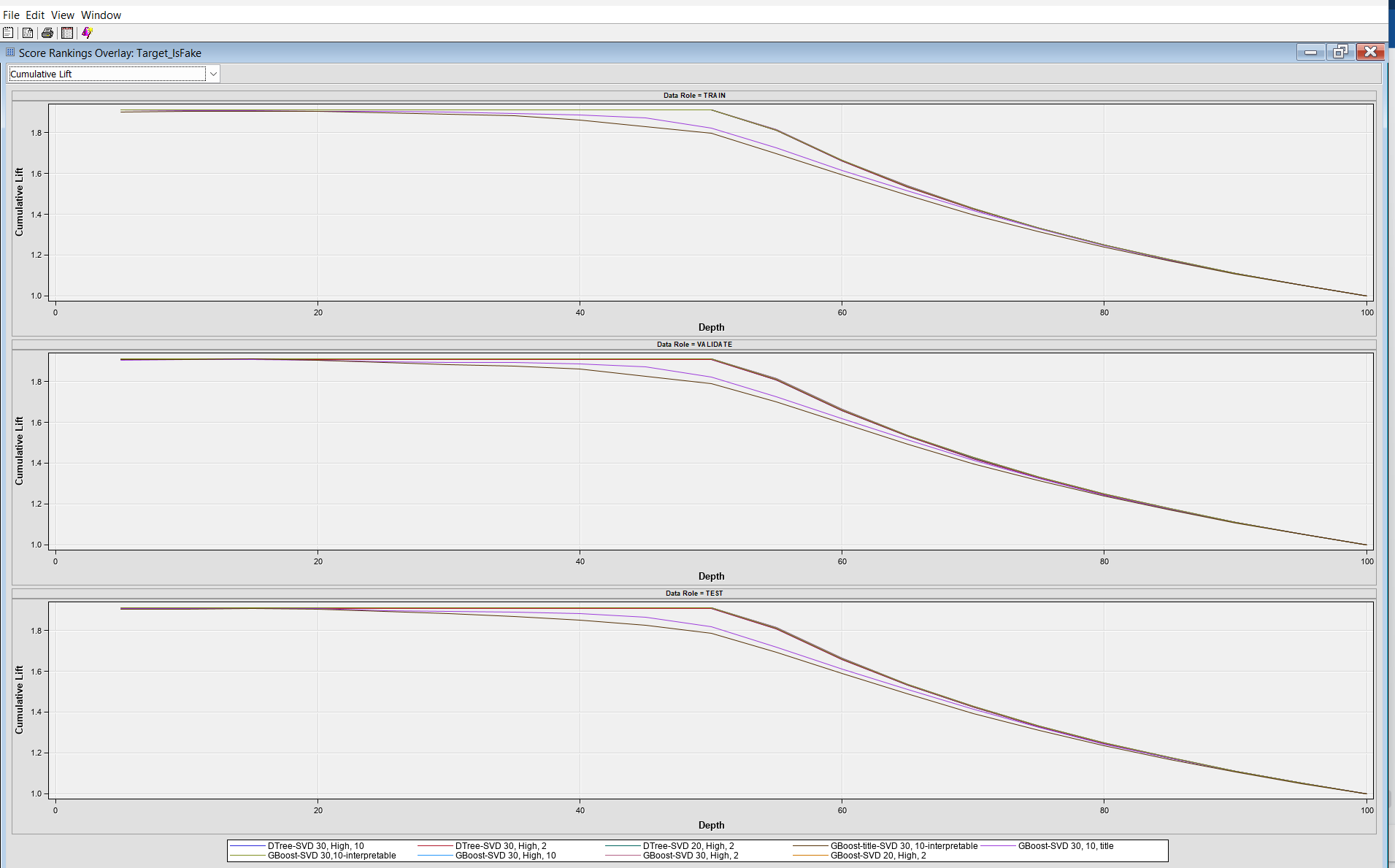
****

**Fit Statistics:**

****

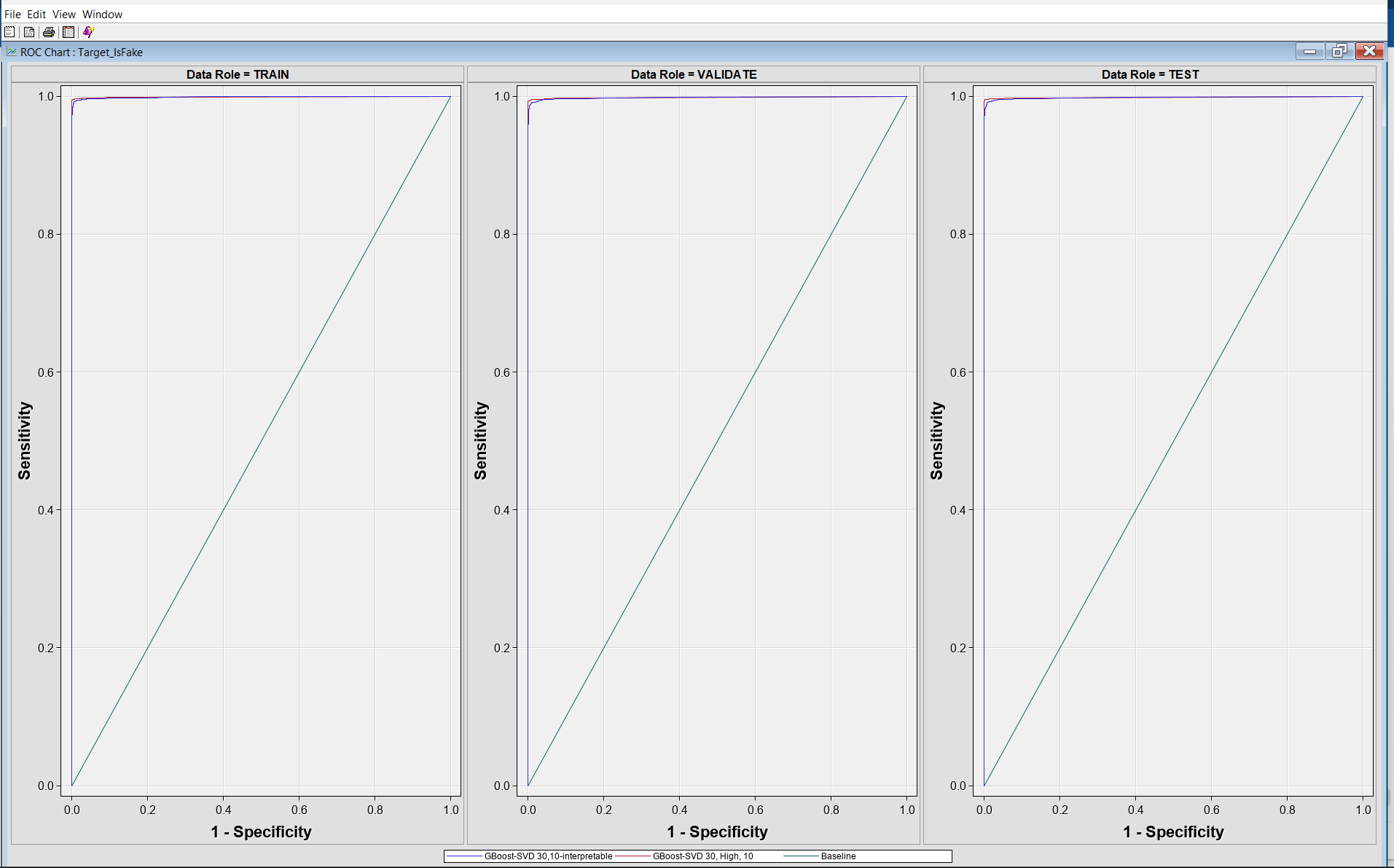
From the model comparison results, Gradient Boosting Model performs better than Decision Tree in terms of misclassification rate, ROC on test partition. With a higher number of SVDs, Gradient Boosting performed better, while the decision tree worked well with a lower number of SVDs. Further the difference between train and validation misclassification rate is a bit higher for decision trees, which signifies that the model might be overfitting. Hence we chose Gradient Boosting over Decision Tree Models. Further, we chose interpretable model as our best model, as it can throw more business insights and gain more trust from the business.

**Score ranking overlay:**

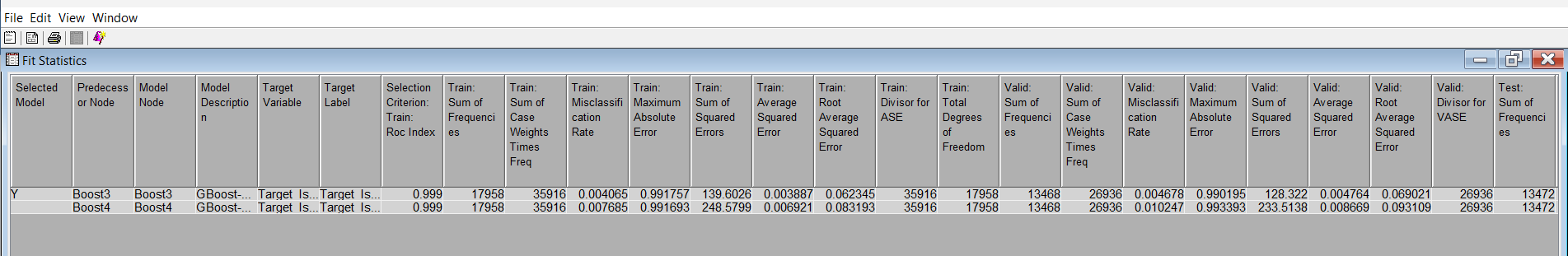


**Model Comparison results (2):**

**ROC Curve:**

****

**Fit Statistics:**

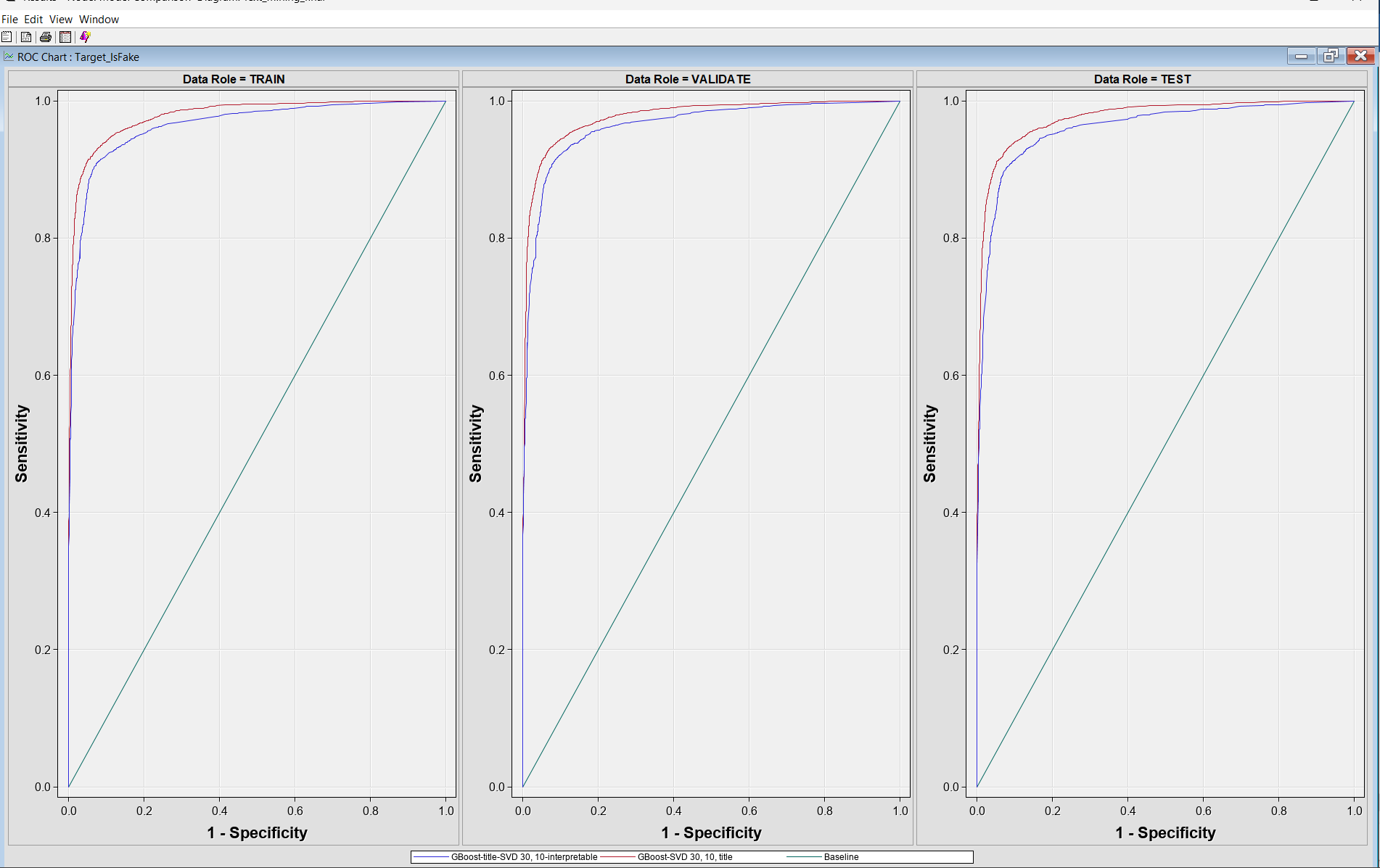
****

**Score Rankings Overlay:**

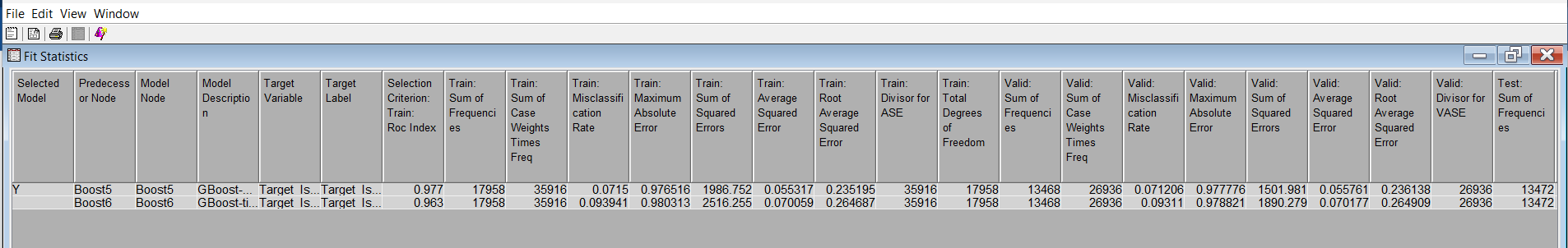
****

**Model Comparison(3):**

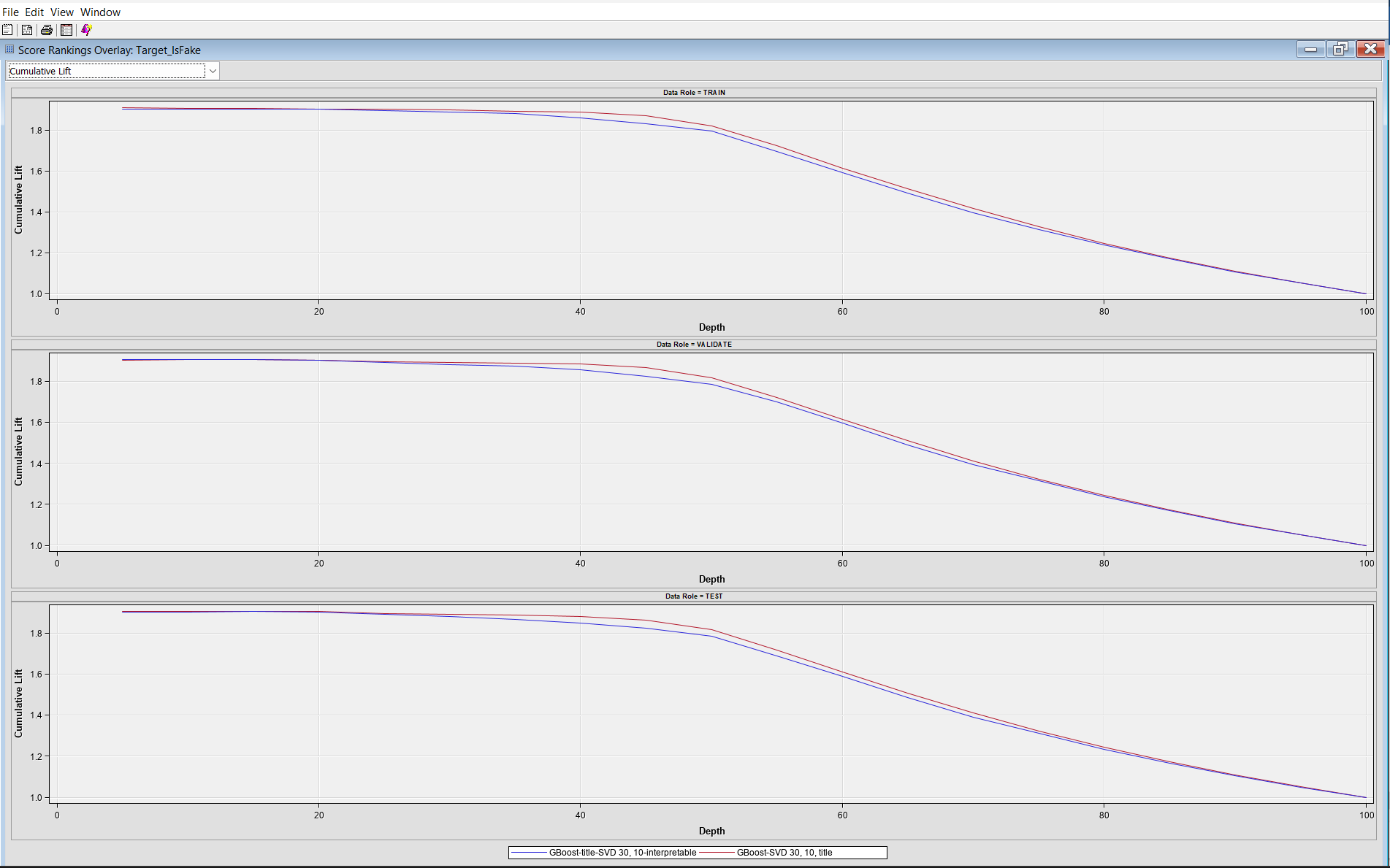
**ROC Curve:**

****

**Fit Statistics:**

****

**Score Ranking:**

****

**Final Model:**

****

#### Recommendations

Mimicking Legitimate Sources:

* Fake news articles may intentionally use the names of reputable sources to lend an air of credibility to their content. Businesses should implement advanced source analysis techniques, combining metadata features like source reputation and domain analysis, to distinguish between genuine and fake articles effectively.

Dynamic Feature Engineering:

* Static keyword-based features may not capture the evolving tactics used by fake news creators. To stay ahead of evolving tactics, businesses should regularly update their detection models with new, relevant features. This dynamic approach ensures continued effectiveness in identifying emerging patterns.

Human-in-the-Loop Validation:

* Certain words or phrases may have legitimate uses in news articles but can also be exploited in fake news. Establish a human-in-the-loop validation mechanism where flagged articles are reviewed by human experts. This human touch is crucial for capturing nuances that automated systems might miss.

Explainability and Interpretability:

* Choose models that provide interpretability and can explain the basis of their predictions. This helps build trust and transparency in the decision-making process, enabling better collaboration and understanding.

Continuous Monitoring and Adaptation:

* Establish a system for continuous monitoring and adaptation. The business should be proactive in updating detection models to counter new patterns and tactics employed by those attempting to spread misinformation.

User Feedback Mechanism:

* Implement a user feedback mechanism that allows users to report potentially false information. This provides valuable data for model improvement and refinement, creating a more responsive and accurate system.

**Conclusion:**Based on our experiments conducted so far, the gradient boosting model performed better than the decision tree in terms of performance measures like Roc AUC, Misclassification rate on the Test partition. The Decision Tree was slightly overfitting, which is why the decision tree that had a lower number of SVD features as input outperformed the one with a higher number of SVDs. The decision whether or not to choose an interpretable model over the usual Gradient Boosting is purely a business choice. An interpretable model in this context can gain more trust and throw useful insights into why the model classifies a particular news article to be fake. Further, the model built on the ‘title’ column alone underperformed. This is expected, as the ‘title’ column has fewer words and hence carries less information than the ‘text’ column. We could probably mix both ‘title’ and ‘text’ into a single text column and build one single model, but we might lose interpretability and it will be difficult to explain what kind of words in ‘title’ or ‘text’ alone result in it being classified as fake. One other way to achieve this without compromising on the interpretability is by building a stacked model that aggregates output from individual models built on ‘title’ or ‘text’ alone and classifies an article accordingly as fake or real. This will add to the future scope of this project.