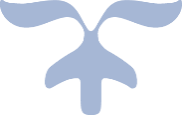
****

**OPIM 5671 – Data Mining and Business Intelligence**



**RESUME ANALYZER**





**Group 7:**

1. **Urvashi Bhurase**
2. **Sai Bharghav Chollangi**
3. **Anusha Derangla**
4. **Vishista Madhi Reddy**
5. **Tharun Mahesh Palla**

Table of Contents

[**Introduction** 3](#_30j0zll)

[**Background and Business Objective:** 3](#_3znysh7)

[**Data Description** 4](#_2et92p0)

[**Data Cleaning** 6](#_3dy6vkm)

[**Text Preparation** 7](#_1t3h5sf)

[**Text Clustering – Unsupervised Learning** 9](#_4d34og8)

[**Interpretable Model** 26](#_2s8eyo1)

[**Model Comparison** 26](#_17dp8vu)

[**Best Model** 26](#_3rdcrjn)

[**Insights** 26](#_26in1rg)

[**Business Value** 26](#_lnxbz9)

[**Conclusion** 26](#_35nkun2)

[**Future Work** 26](#_1ksv4uv)

[**References** 26](#_44sinio)

# **Executive Summary**

The objective of our SAS Enterprise Miner project was to create a resume analyzer that could reduce the time and costs associated with manual resume screening, while also increasing the accuracy and effectiveness of the recruitment process. Our approach involved using models to analyze and categorize large volumes of resumes in a fraction of the time it would take a human recruiter. To achieve this objective, we chose four categories - Advocate, Business Development, IT, and Designer - and worked on 100 resumes for each category. We then performed an analysis of the resumes using SAS Enterprise Miner, which also included the use of another dataset containing job descriptions to better understand what words were important in each description.

We created four models based on different term weights in SAS, including supervised and unsupervised models with four clusters, using job category as the target variable. The four models were Neural Networks, Decision Tree, Gradient Boosting, and Regression. Our best model was Gradient Boosting with Mutual Information, which had a misclassification rate of only 0.00361. Additionally, we created an interpretable model of the Decision Tree to understand how accurate our models were.

Our project has significant potential benefits for the recruitment process, as it enables the analysis and categorization of resumes in a much shorter time than traditional manual methods. By reducing the time and costs associated with manual screening, we believe our project could significantly improve the recruitment process and increase its accuracy.

Overall, we are confident that our project has demonstrated the potential of SAS Enterprise Miner in resume analysis and has provided a valuable tool for improving the recruitment process.

# **Introduction**

In today's job market, companies are inundated with a flood of resumes from potential candidates. To manage this overload of information, companies often use automated systems to filter and prioritize incoming resumes based on specific criteria. However, traditional Applicant Tracking Systems (ATS) can suffer from limitations, such as high false-negative rates and difficulty in adapting to changes in job requirements. To address these issues, we propose a novel approach to resume analysis using text mining techniques. In this paper, we present the development of a Resume Analyzer that categorizes resumes into four topics, namely IT, Designer, Business Development, and Advocate. The analyzer is trained using multiple machine learning models, with a decision tree model achieving the best performance based on misclassification rate. We demonstrate the effectiveness of our approach in reducing costs and improving the quality of candidate selection, while also providing insights into the key factors that distinguish each job category. Our paper presents a detailed description of the methodology used in our model, as well as an analysis of the results and potential areas for future research.

# **Business Objective**

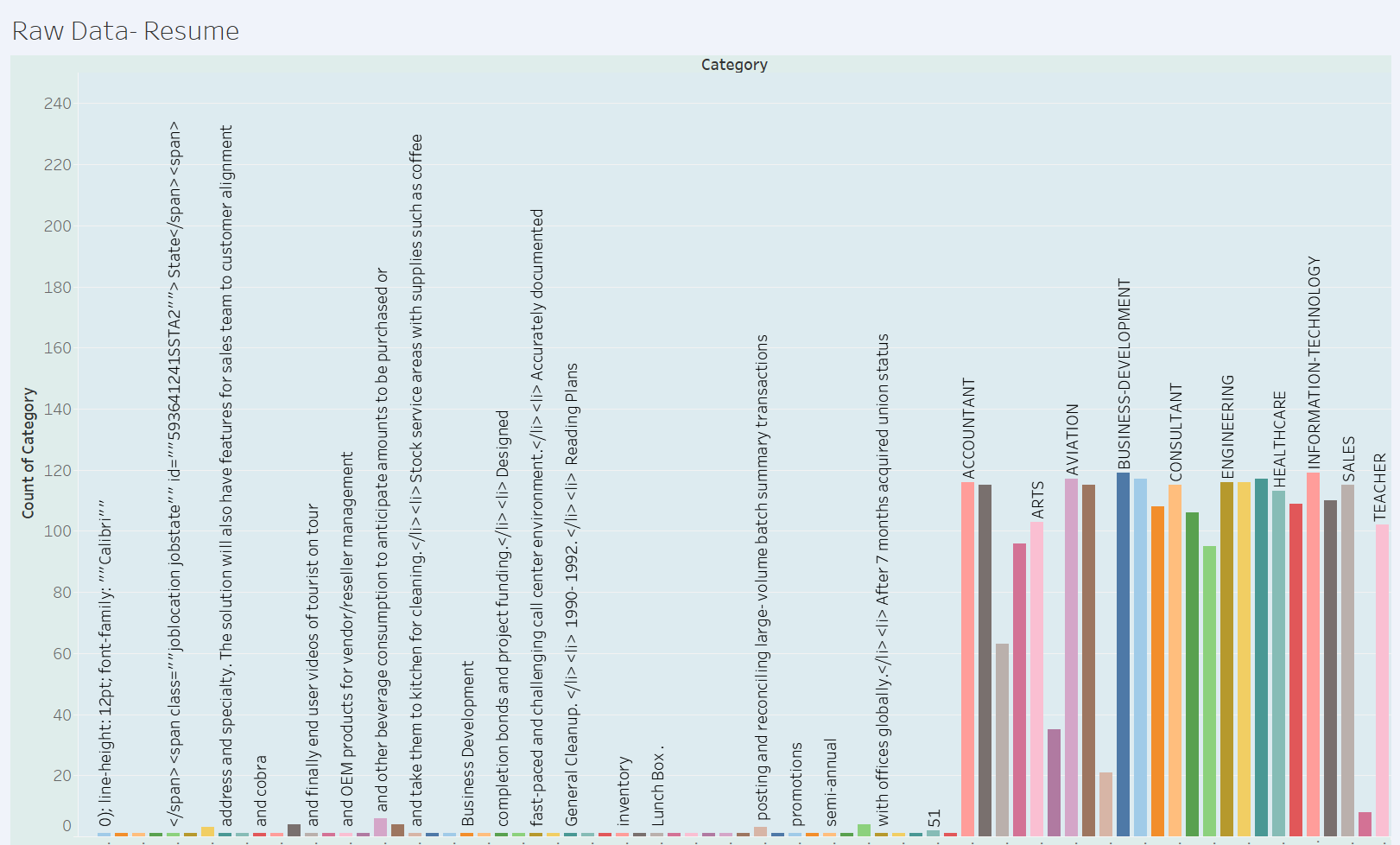
The business objective of this project is to improve the efficiency and effectiveness of the recruitment process by automating the initial screening of resumes and identifying the best-fit candidates for each job category. This could reduce the time and costs associated with manual resume screening and increase the accuracy and effectiveness of the recruitment process. By using these models, it is possible to analyze and categorize large volumes of resumes in a fraction of the time it would take a human recruiter.

This analysis can benefit job seekers by providing insights into the skills and keywords that are in demand for each job category. Candidates can use this information to tailor their resumes and applications to match the specific requirements of the job, increasing their chances of getting hired. Additionally, this project can help organizations better understand the skills and experiences that candidates possess and identify areas where they may need to provide additional training or development.

Moreover, the data collected through this project can provide valuable insights into the job market, helping companies stay competitive by identifying important skills and experiences. Organizations can use this information to adjust their hiring strategies and focus on recruiting candidates with the most in-demand skills and experience. Overall, the project will help streamline the recruitment process, reduce costs, and provide valuable insights into the job market and the skills and experience in demand for each job category.

# **Data Description**

Our dataset, which we obtained from Kaggle, contains more than 3000 resumes distributed among 24 distinct job categories. These categories encompass a broad range of occupations, including human resources, design, information technology, advocacy, agriculture, sales, and automobiles. Each row in the dataset is uniquely identified by an ID and includes relevant information, such as the job title in the "resume\_str" field and the actual resume text in the "resume\_html" field. The resumes are also assigned to specific job categories, which is a critical feature of the dataset.



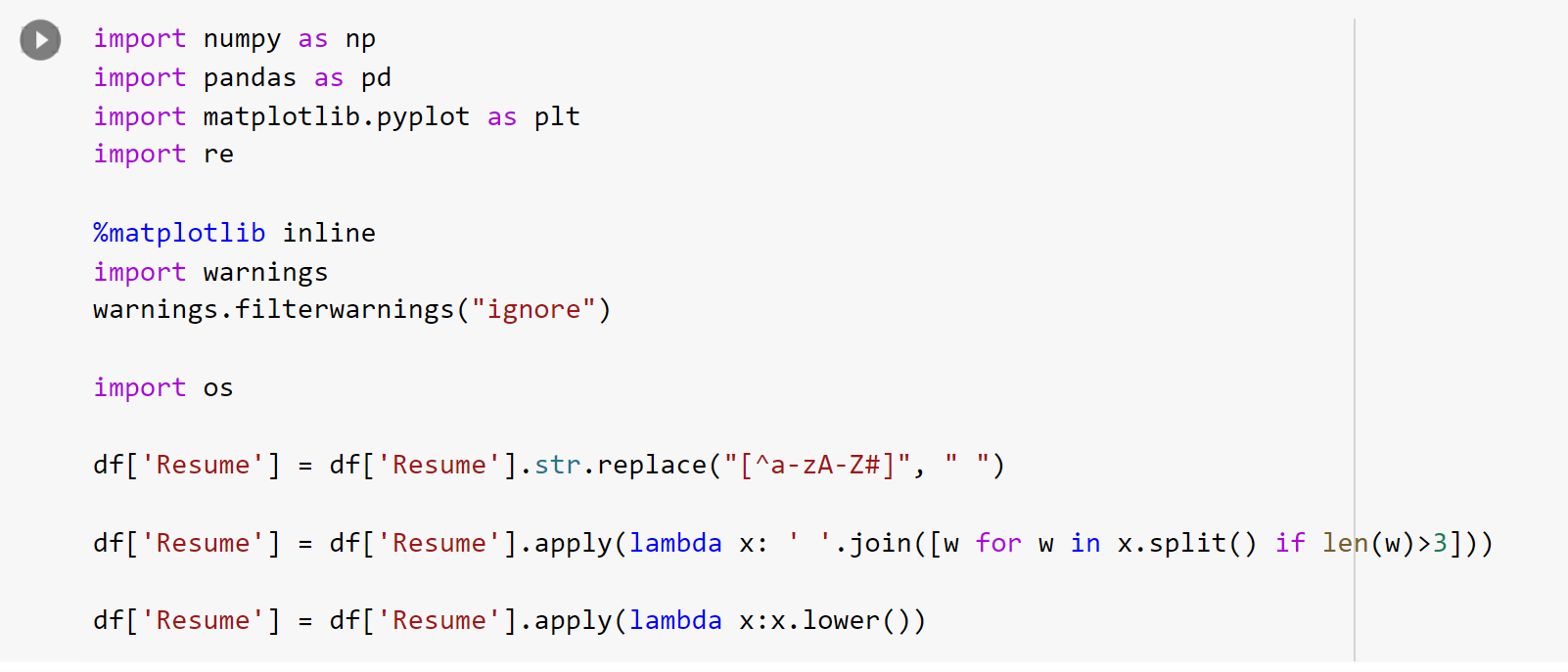
We have narrowed down our analysis to 100 resumes, categorized into four distinct groups, from the original dataset of 3000 resumes. Here is a screenshot of the count of resumes and categories that we have selected for our analysis:

Graphical user interface, application, Word

Description automatically generated

# **Data Preprocessing**

Data preprocessing is a vital step in text mining, as it helps to reduce noise, improve consistency, extract useful features, and reduce data redundancy. In our text mining project, data preprocessing was a crucial step in preparing our dataset for analysis. We began by removing any unnecessary rows and columns and removing any blank cells in Excel. We then selected 100 rows for each job category to obtain a manageable dataset for analysis.



Following the initial Excel cleaning, we proceeded to use Python for further data cleaning and preprocessing. The code provided above is one of the preprocessing steps we used in Python. Firstly, we removed any non-alphabetic characters using regular expressions to replace all non-alphabetic characters with spaces. This step helped to ensure that our text data was consistent and would not be affected by any symbols or numbers that could skew the results.

Next, we used lambda functions to remove any words that were shorter than three characters. We found that many words with only one or two characters appeared frequently in the resumes, but they were not relevant to the analysis. Removing them helped to reduce the size of our dataset and increase the accuracy of our analysis. At last, we converted all the text to lowercase to ensure that our analysis would not be affected by the presence of capital letters. This step also helped to standardize our text data and eliminate any potential inconsistencies.

Our data preprocessing involved both Excel and Python cleaning to ensure that our dataset was clean and consistent for analysis. The cleaning involved removing unnecessary rows and columns, removing blank cells, and cleaning the text data using regular expressions and lambda functions. These steps helped to ensure that our text data was accurate, consistent, and ready for analysis.

# **Text Preparation**

To prepare text data form machine learning models, we first passed the cleaned imported data through the data partition node. We have partitioned our data into 70% training, 20% validation and 10% test data. In our project, we have a dataset of resumes that are categorized into four major categories: Advocate, Business Development, Designer, and Information Technology. The distribution of resumes across these categories is crucial to building an accurate and reliable model. Stratified sampling was chosen because it ensures that each category is represented in the training, validation, and testing datasets in proportion to its representation in the overall dataset.

Stratified sampling guarantees that each category is represented in the sample with the same proportion as it appears in the population, thus avoiding sampling bias that could occur with other sampling techniques. As a result, this method ensures that the model we develop will be well-suited for classifying resumes across all four categories, not just one or two. By utilizing stratified sampling, we can increase the model's accuracy and reduce the likelihood of misclassification, making it more reliable and effective for real-world applications.

Table

Description automatically generated

After Partitioning the data, we performed text parsing to convert unstructured text data into a structured format that can be analyzed. In this project, the text parsing step is important because it helps in removing extraneous or misleading data such as punctuation marks, special characters, and other noise that can negatively impact the accuracy of the text mining models.

During the text parsing step, we can also remove stop words, which are common words such as "the", "a", and "an" that do not carry any significant meaning in the context of text mining. By removing stop words, we can focus on the more meaningful terms that are relevant to our analysis, such as job titles, skills, and experience.

We customized the stop list, which is a list of words that are excluded from the text analysis because they are deemed irrelevant or do not add value to the analysis. By removing certain terms from the stop list, we can improve the quality of our analysis by ensuring that we are focusing on the most important terms and concepts related to our project for e.g. Indeed, associated, willing, performed etc.

Table

Description automatically generated

After parsing the data, this is passed to the text filter node**.** Text filtering allows us to apply different combinations of frequency weights and term weights to our data. Frequency weights consider the frequency of occurrence of each term in the document, while term weights measure the importance of each term relative to the entire document set. By experimenting with different combinations of frequency weights and term weights, we can identify which combination produces the most accurate and meaningful results.

Below are the Frequency and Term Weightings we tried in our project:

* **Frequency Weight**: Log, **Term Weight**: None
* **Frequency Weight**: Log, **Term Weight**: Entropy
* **Frequency Weight**: Log, **Term Weight**: IDF
* **Frequency Weight**: Log, **Term Weight**: Mutual Information

# **Text Clustering – Unsupervised Learning**

Text Clustering node is used for clustering unstructured text data. It takes in a collection of text documents and clusters them based on the similarity of their content. The node uses a combination of statistical techniques such as Singular Value Decomposition (SVD), Expectation-Maximization (EM) algorithm, and probabilistic models to cluster the documents.

Each text filter node was attached to a text cluster node, with SVD (Singular Value Decomposition) Resolution set as high and a maximum number of SVD dimensions as 10.

By using SVD resolution, we were able to reduce the dimensionality of the data and represent it in a lower-dimensional space. This helped to improve the efficiency of the clustering process and avoid overfitting. We defined the number of clusters as 4, as our dataset had four major categories to categorize each resume into.

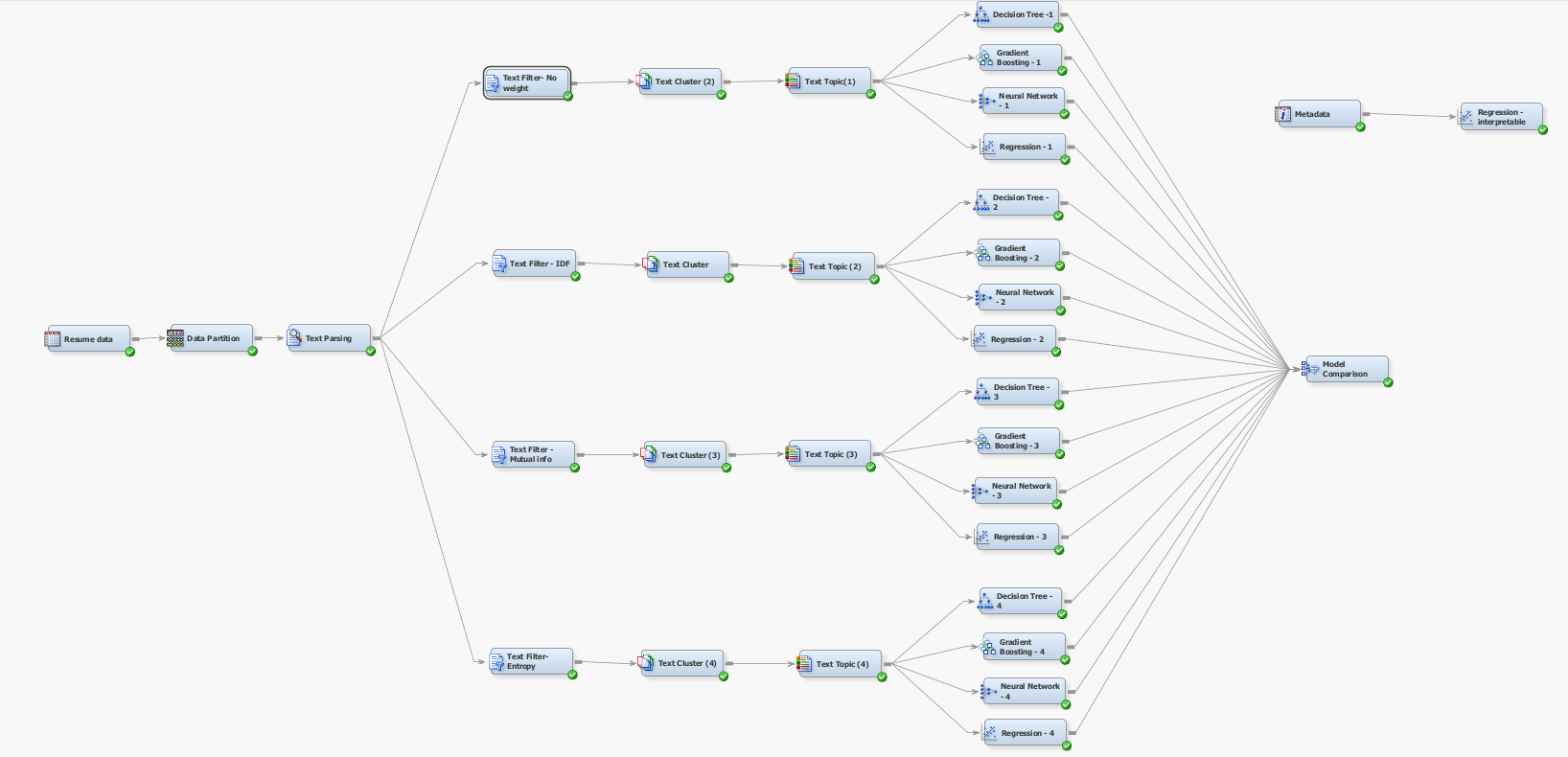
We used the Expectation-Maximization cluster algorithm, which is a commonly used algorithm for text clustering, and used the TextCluster\_prob variables created by the node as input variables to increase the interpretability of our model. Finally, we kept the number of descriptive terms for each resume to 30. The algorithm assigns each document to a cluster based on the probability of it belonging to that cluster. The cluster assignments are updated iteratively until convergence.

Finally, the node produces a set of output tables containing information about the clusters such as the number of documents in each cluster, the most significant terms in each cluster, and the probability of each document belonging to each cluster.

**Supervised ML Modeling**

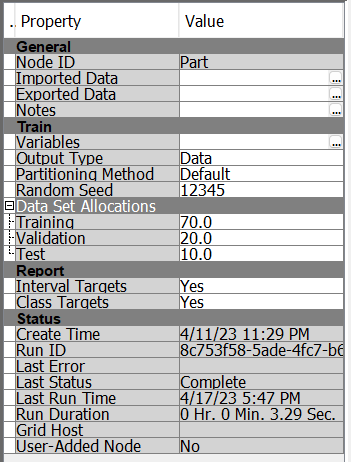
**Text mining capabilities are provided by the SAS Enterprise software suite's Text** Analytics module.

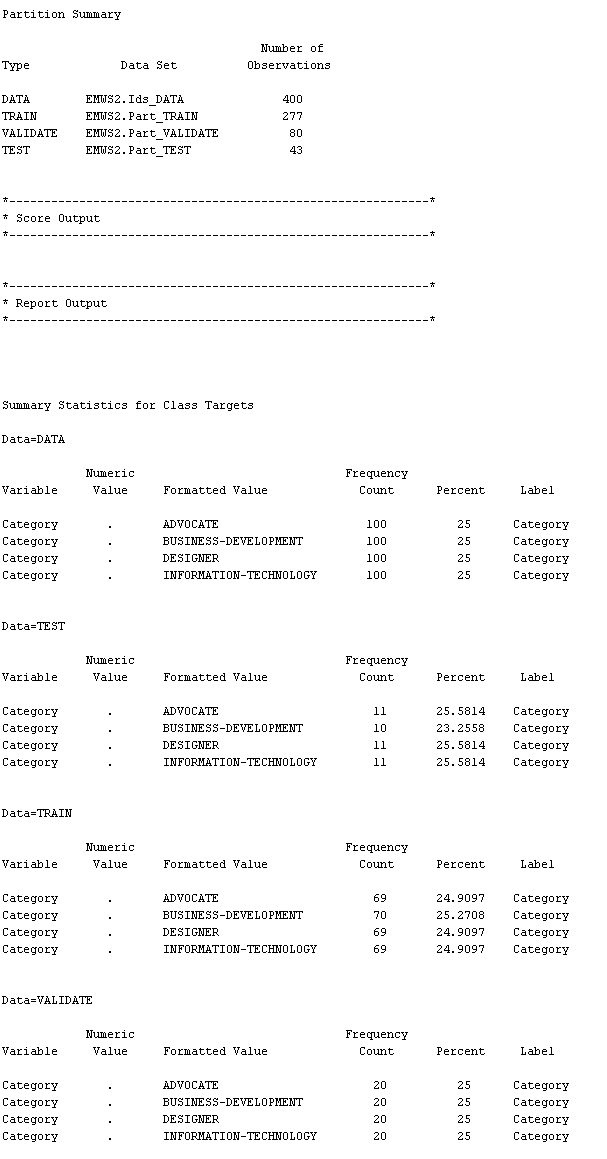
**Below is the diagram of text mining for the resume dataset.**

****

We have the resume data which we have partitioned into training 70, validation 20 and test as 10.

We use the training set to train the model on how to classify the target variable based on the input features. The model's parameters are adjusted using the validation set, and its performance is evaluated when applied to new data. Finally, the model's performance is assessed using the test set.





We have included text parsing node to the data partition, With the help of the Text Parsing node in SAS Enterprise Miner, we can extract structured data from unstructured text data and find insights and patterns that manual analysis might miss.

Table

Description automatically generated

As shown in the example below, each term plays a part in the text parsing process.Table

Description automatically generated

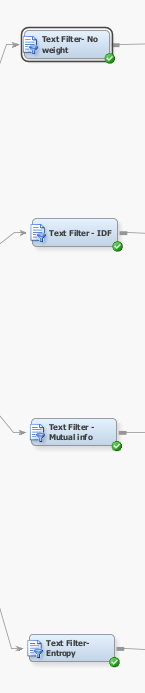
Later we tried to change term weight. The value or significance of a term (i.e., a word or a phrase) in a text document is referred to as term weight in SAS. In text mining and natural language processing applications, the term weight is a numerical value that reflects the relative importance of the term in the document or corpus.

In SAS, term weight can be calculated in a number of ways, including IDF, No weight, Mutual information, Entropy to check which one is performing better.

Graphical user interface, application

Description automatically generated with medium confidence

We may choose the term weights based on our preferences and determine which one truly works better by looking at the term weight as seen in the screenshot above.

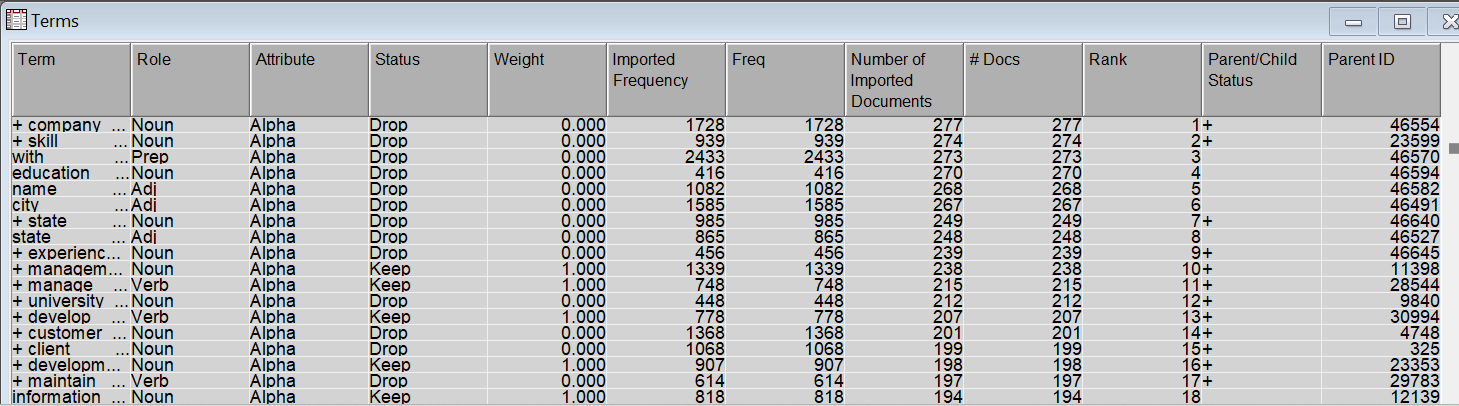


A term's weighting system called **Inverse Document Frequency (IDF**) is utilized in SAS Enterprise Miner to gauge a term's significance inside a corpus of documents.

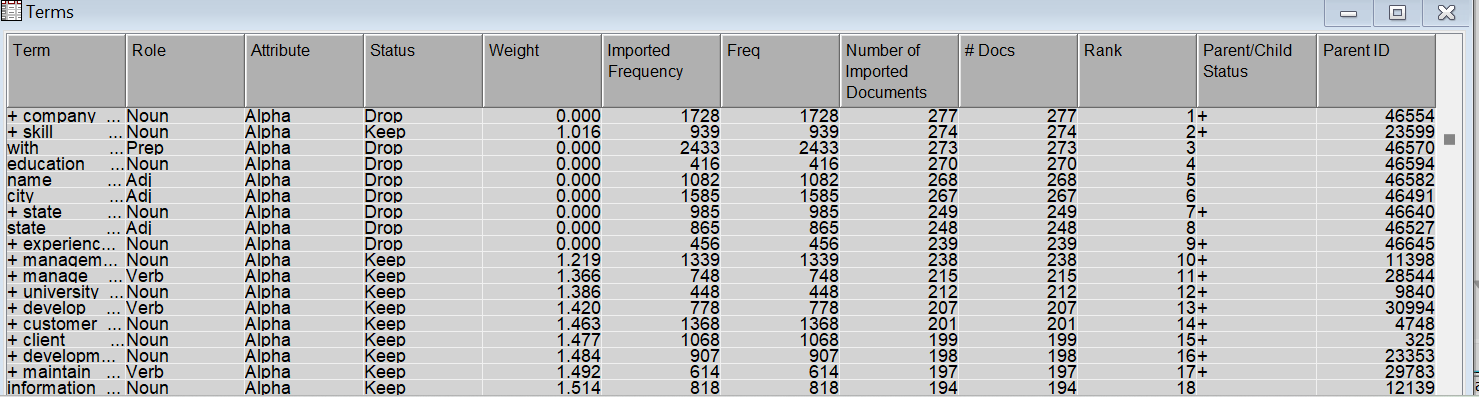
SAS Enterprise Miner uses the term-weighting scheme known as "**Mutual Information**" to assess the level of relationship between a term and a class label inside a corpus of documents.

**Entropy** is a term-weighting method used in SAS Enterprise Miner to assess the degree of a term's unpredictability in relation to a class label in a corpus of documents.

**Below is the screenshot with no weights:**



**Below is the screenshot with term weight as IDF**

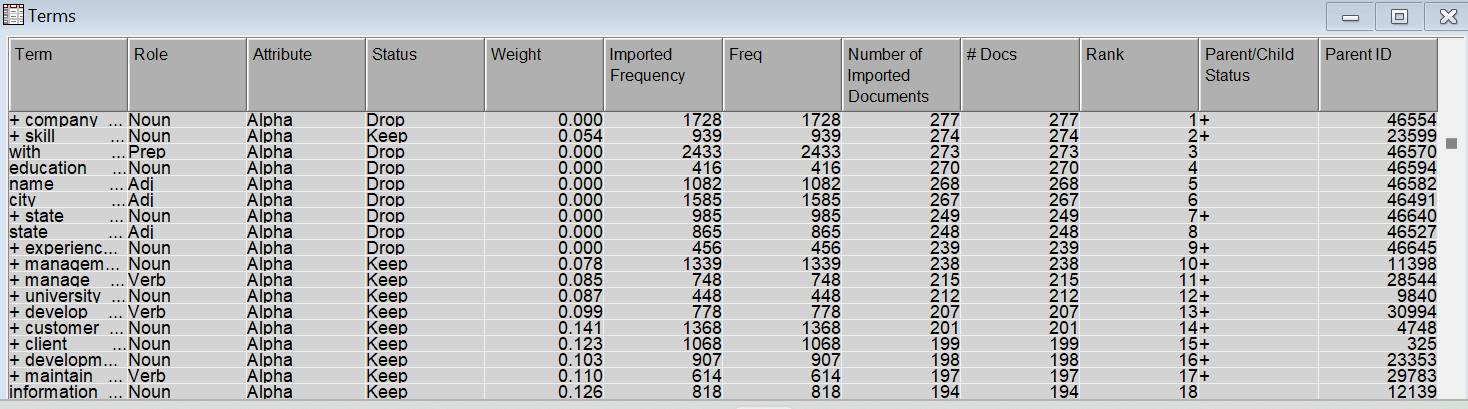


**Below is the screenshot with Mutual info**

Table

Description automatically generated

**Below is the screenshot with Entropy**

****

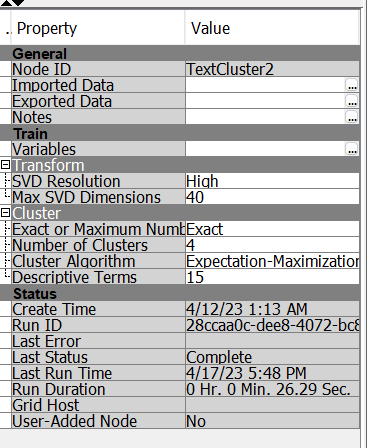
We can see when we changed the term weight, the weight kept on changing as per the above pictures.

Text clustering: It is the process of assembling related texts or documents into clusters based on their subject matter, tone, or other attributes. For purposes including information retrieval, document categorization, and topic modeling, text clustering aims to unearth hidden patterns or structures in massive collections of text data. Text clustering is the process of finding similarities and differences across texts and organizing them into useful groups using methods like statistical analysis, machine learning, and natural language processing.

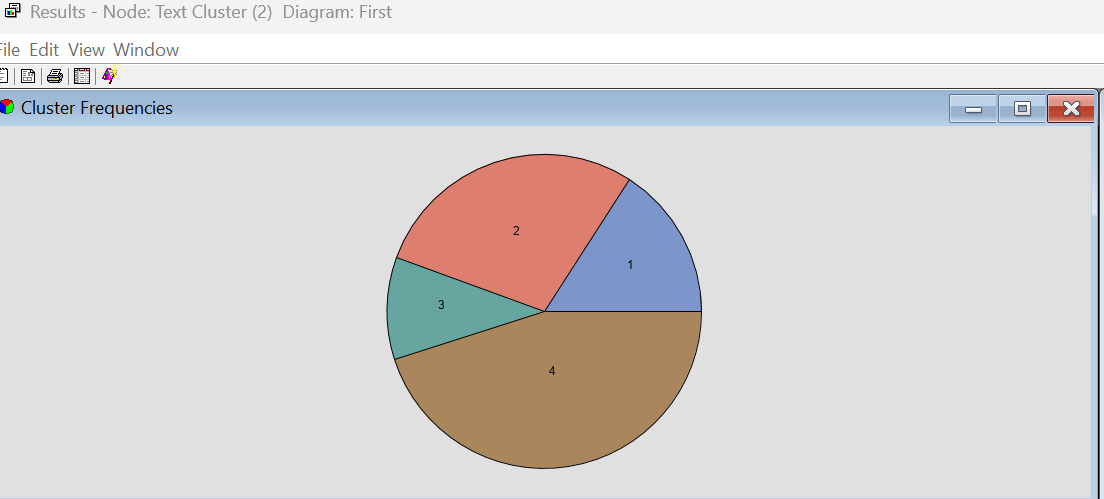
As we have 4 resume categories:

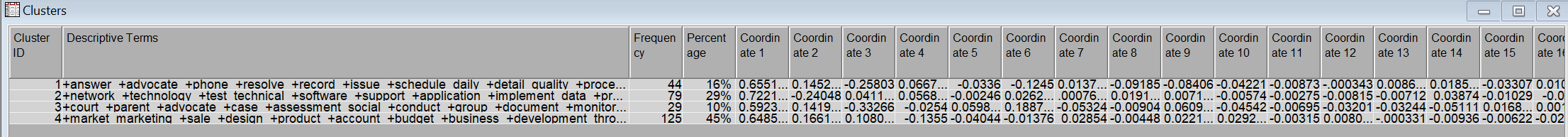
1. Designer
2. Advocate
3. Business- development
4. Information technology

We created exactly 4 clusters using text cluster node



Cluster 2 with term weight none

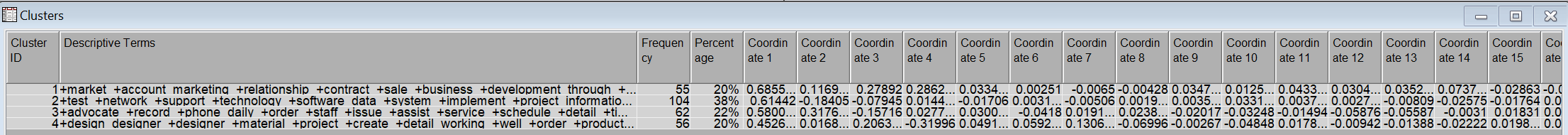




Cluster 1 with term weight Inverse frequency document

Chart, pie chart

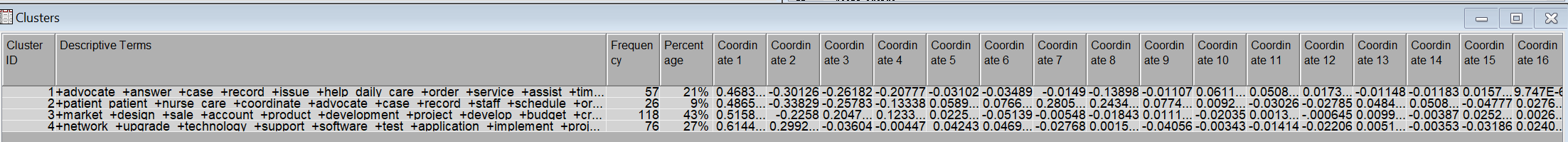
Description automatically generated



Cluster 3 with term weight mutual info

Chart, pie chart

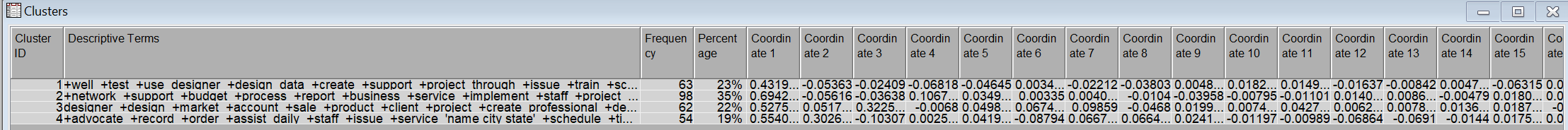
Description automatically generated



Cluster 4 with term weight entropy

Chart, pie chart

Description automatically generated

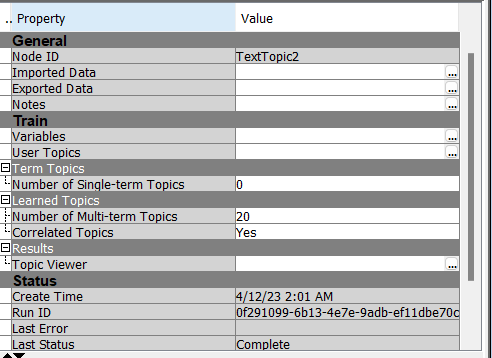


After attaching Text Cluster nodes to the Text Filter node, we have attached each Text Cluster with a respective Text Topic node.

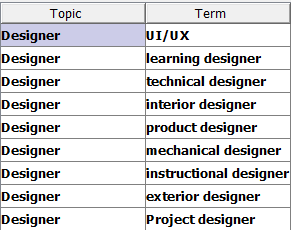
**Text Topic**

In the Text Topic node, we have specified the number of single term topics = 0 and number of multi-term topics = 20 keeping the rest of the settings as default in the properties panel of the node. In our Text Topic node, we encountered a challenge with unclear word generation from the initial model used to classify resumes. To address this issue, we utilized our second dataset to create User topics manually for each job category. By selecting 10 job titles as topics and creating subtopics with the job title as the main topic and the associated skills as terms under the subtopic, we were able to incorporate these User topics into our models. This allowed us to improve the overall accuracy of our analysis by more effectively categorizing resumes based on their relevant job skills and requirements.

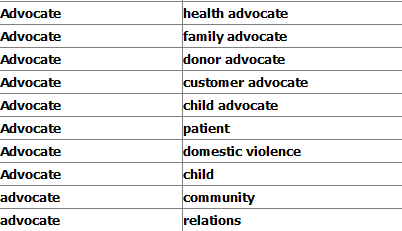
Incorporating User topics into our analysis has had a positive impact on the number of documents by topic and has improved our overall model's accuracy. By manually creating User topics for each job category and incorporating them into our analysis, we were able to more accurately classify resumes based on their relevant skills and experience.



Job titles as terms based on categories we chose:

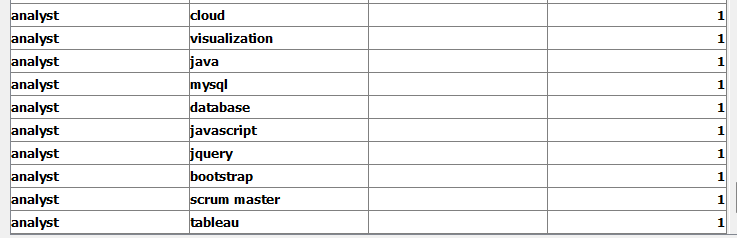
 

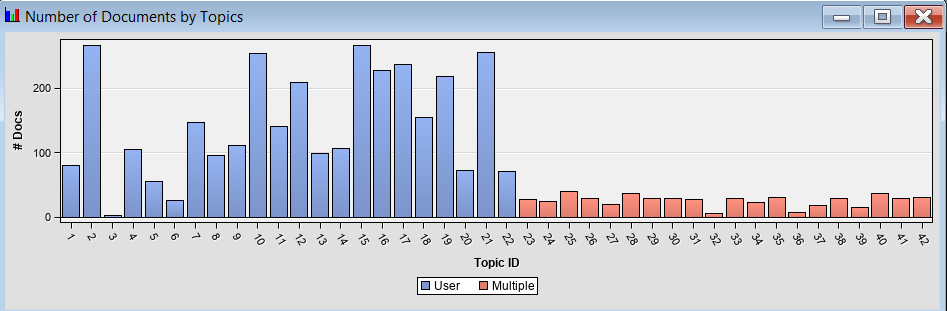




Skills for each Job title:

Example:

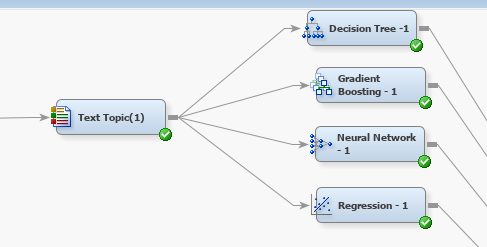


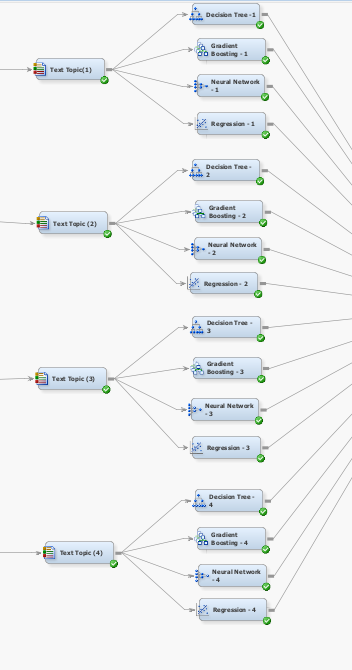


By the Graph, we were able to understand that the user topics we inhibited into the mode had more weightage and would help in better classifying the terms as per our goals,

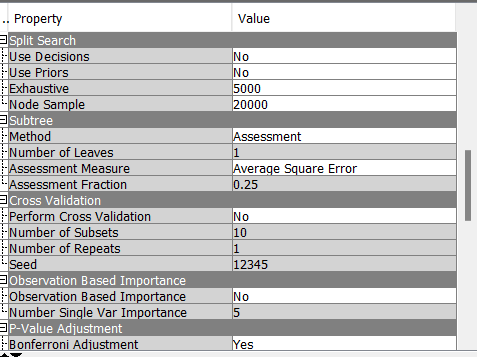
**Modelling:**

Post setting up the basic data flow we decided to experiment with four different models namely, Decision Tree, Gradient Boosting, Neural Network, Regression. Following the plan each text topic node was attached to four modes each corresponding to a model (i.e., Decision Tree, Gradient Boosting, Neural Network, Regression).

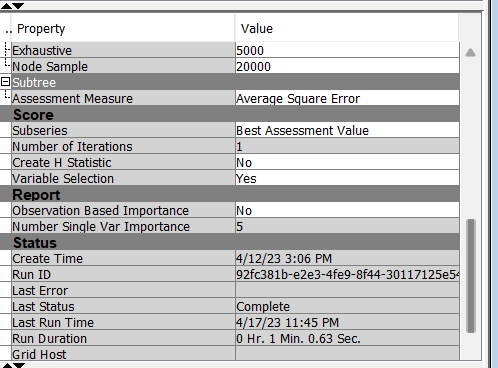




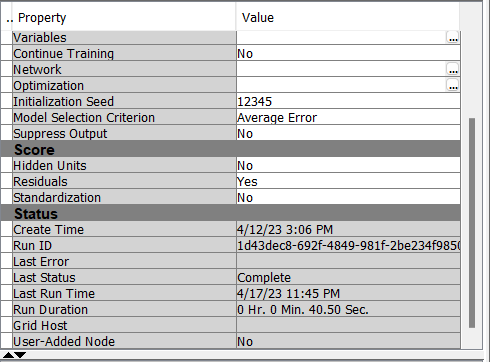
**Decision Tree:** The assessment method for all the decision tree nodes was set to Average Square Error keeping the rest of the settings as default.



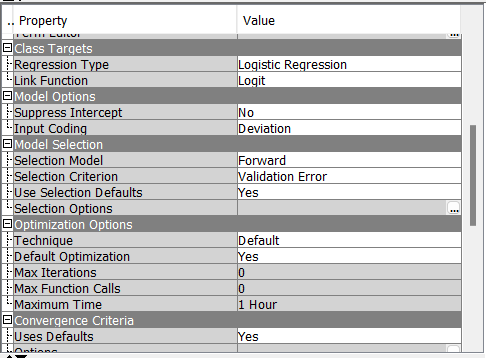
**Gradient Boosting:** Similar to the Decision Tree node the Gradient Boosting node was also set to Average Square Error as it often produces better results with trees. The rest of the settings were kept to default.



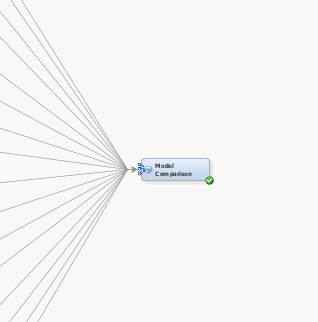
**Neural Network:** In the Neural Network node we have defined our Model Selection Criterion as Average Error keeping the rest of the setting as default.



**Regression:** In the Regression node we have specified the Forward Logistic Regression Model with selection criteria as Validation error keeping rest of the settings as default.



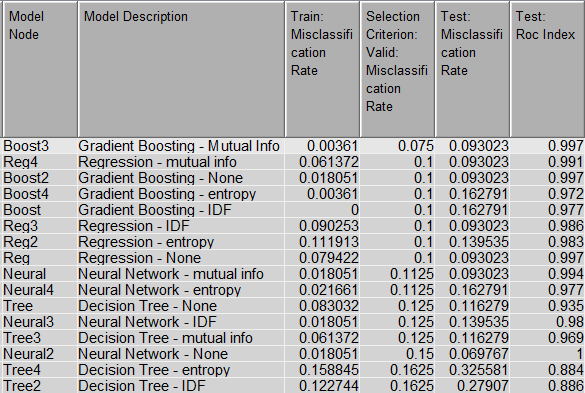
Finally, all the 16 nodes of four models from four respective text topic nodes were attached to a Model Comparison node to check the accuracy of each model.

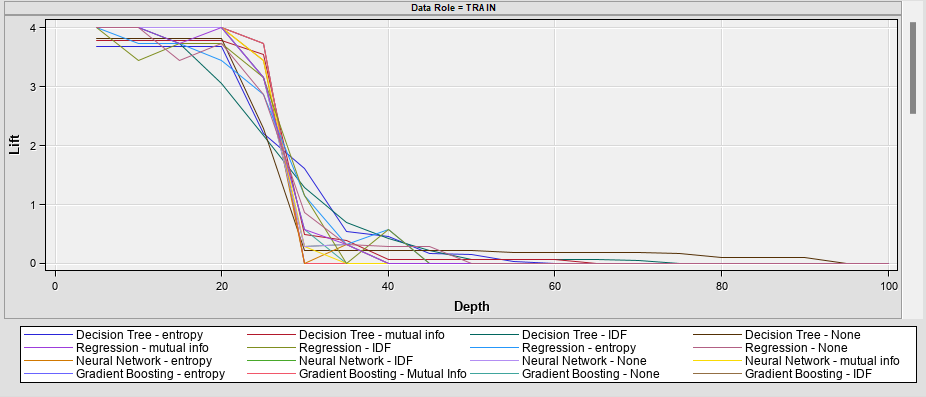


# **Interpretable Model**

**Model Comparison**

We have 16 models in total for each of the four term weight settings namely, None, IDF , Entropy and Mutual Information. Below are the Fit Statistics obtained from the Model Comparison.





**Best Model**

So the best model as per the fit statistics was found to be Gradient Boosting model with term weight set as Mutual Info giving us the least misclassification rate of 0.075 on the validation set and ROC Index of 0.997 on the test set.

**Insights**

The text mining model we developed was aimed at identifying the key skills and job titles present in the resumes of four categories: Advocate, Business Development, Designer, and Information Technology. The model generated several insights that can be useful for recruiters and hiring managers:

1. **Skills and job titles**: The model identified the most frequent skills and job titles in each category. For example, in the Advocate category, the most common skills included legal research, litigation, and legal writing, while job titles included attorney and paralegal. In the Business Development category, the most common skills were sales, business development, and marketing, while job titles included sales executive and marketing manager. Similarly, for the Designer category, the most common skills included graphic design, user experience design, and web design, while job titles included graphic designer and web designer. In the Information Technology category, the most common skills were programming, software development, and system administration, while job titles included software developer and system administrator.
2. **Differences between categories**: The model identified significant differences between the categories in terms of the skills and job titles present in the resumes. For example, the Advocate category had a greater focus on legal skills and job titles, while the Business Development category had a greater focus on sales and marketing skills and job titles.
3. **Clustering**: The model also clustered the resumes into four categories, which aligned with the categories we had defined. This clustering can help in the automatic classification of resumes based on their content.

Overall, the insights from the model can help recruiters and hiring managers identify the most suitable candidates for a given job role based on their skills and job titles and can also help in the automatic classification of resumes based on their content.

**Business Value**

This project provides significant business value for both recruiters and applicants in the following ways.

**For Recruiters:**

1. **Improved Candidate Screening**: The model helps in screening resumes more effectively, reducing the chances of missing out on potential candidates that might be overlooked using traditional manual screening methods. The model can classify resumes based on the job requirements, which helps recruiters in shortlisting suitable candidates. The model can also filter out resumes that do not meet the job requirements, reducing the time spent in screening unsuitable resumes. This results in improved candidate screening, saving recruiters time and resources.
2. **Reduced Time-to-Hire**: With the model's ability to classify resumes more accurately and quickly, recruiters can reduce the time-to-hire for open positions. The traditional manual screening process can be time-consuming, and the model can streamline the process by filtering out unsuitable resumes, reducing the time recruiters spend on screening. This will result in faster hiring decisions and increased productivity for the recruitment team.
3. **Customizable**: The model can be customized according to the job requirements, allowing recruiters to add new topics and keywords based on their specific hiring needs. Recruiters can also provide weightage to the topics and terms, which means that the model will consider the more important topics and keywords with a higher weightage. This customization provides recruiters with flexibility in their hiring process.
4. **User Interactive**: The model's interactive nature allows recruiters to visualize the clusters and understand which topics are being given more weightage. This can help recruiters to fine-tune the model's output, improving the accuracy of the classification. The interactive nature of the model makes it easy for recruiters to use and understand, which leads to better decision making.

**For Applicants:**

1. **Resume Weight**: The model can help applicants understand the weight of their resume, i.e., which skills and experience are more important for a particular job. Applicants can use this knowledge to tailor their resumes and improve their chances of being shortlisted for interviews.
2. **Filter Job Descriptions based on resume**: Applicants can use the model to filter job descriptions based on their resume, reducing the time spent on searching for suitable job postings. This also helps applicants to apply for jobs that match their skills and experience, increasing their chances of being hired.
3. **Understanding what skills are important in the market**: The model can help applicants understand what skills are in demand in the job market. By analyzing the term weights and cluster probabilities, applicants can identify the skills that are important to recruiters and improve their skill set accordingly. This can help applicants stand out in a competitive job market, improving their chances of getting hired.

**Conclusion**

Based on our analysis, we have chosen Gradient Boosted Trees as the best model for our project. One of the main reasons for selecting this model is that it is easier to interpret by users and provides better accuracy. The interpretability of the model is essential as users should be able to understand how the model works and the importance of different topics or skills. Gradient Boosted Trees is also scalable and can handle large datasets, making it suitable for our project.

We evaluated our models using the misclassification rate, which provides a clear indication of how many resumes the model is misclassifying. This metric can be easily understood by both technical and non-technical stakeholders. Additionally, we also focused on the false negative rate, which gives the percentage of resumes that are classified incorrectly. This metric is especially important for recruiters as they do not want to miss out on potential candidates due to model errors.

In conclusion, our project aimed to develop a text mining model to help recruiters categorize resumes into different job categories. We performed data exploration, cleaning, preparation, and text analysis to develop our model. We used stratified sampling to ensure an equal representation of different categories and performed text filtering and parsing to extract relevant information. We then used text clustering to categorize resumes into different job categories and evaluated our models using the misclassification rate and false negative rate. Our model provides valuable insights for recruiters and can save them time and effort in categorizing resumes, thereby improving their recruitment process. Overall, our project demonstrates the effectiveness of text mining in providing useful insights from unstructured data.

Diagram

Description automatically generated

# **Future Work**

The future scope of this project is quite extensive and can provide a lot of value to recruiters and applicants. One of the primary areas where this project can be improved is by implementing multilingual support. As the job market is becoming increasingly global, recruiters may receive resumes in different languages. By adding support for different languages, this model can be expanded to cater to a wider audience, and hence, provide more value.

Moreover, this project can be extended to add more categories based on requirements. As job descriptions become more specialized, there may be a need to add more categories to classify resumes more accurately. This can be achieved by retraining the model with more data and adding new categories to the training dataset.

Nowadays, online portfolios are becoming more popular among job seekers. As part of the further scope, this project can be extended to scrape data from the applicant's portfolio link and perform classification, providing feasibility to the recruiter. This would make the hiring process more efficient as recruiters can now analyze more information about the applicant, providing a more holistic picture of the candidate.

Another area where this project can be improved is by using a deep learning model like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). These models can handle unstructured data like text and images much more effectively, and hence can provide more accurate results. However, these models are more complex and require a lot more data and computational resources to train.

Finally, this project can be integrated into recruitment software used by companies. By doing so, recruiters can easily analyze resumes and make more informed decisions. Furthermore, by automating the resume screening process, this project can save a lot of time and effort for recruiters, allowing them to focus on other important tasks.

# **References**

1. Yi-Chi Chou, Han-Yen Yu, "Based on the application of AI technology in resume analysis and job recommendation", *2020 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pp.291-296, 2020.
2. Sania Khan, "An efficient human resource management system model using web-based hybrid technique", *Problems and Perspectives in Management*, vol.20, no.2, pp.220, 2022.
3. Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. Bioinformatics, 1(1), 1-12.
4. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge University Press.
5. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1-47.
6. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer.
7. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513-523.
8. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1-2), 1-135.
9. Berry, M. W., Dumais, S. T., & O’Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. SIAM review, 37(4), 573-595.
10. Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks (pp. 45-50).
11. Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 1-27.

# 