ISSN PRINT 2319 1775 Online 2320 7876

Research paper Volume 11, Iss 10, 2022

Dimensionality Reduction for Brazilian Business Descriptions

Venkateswarlu B[1*], Dr Somasekhar Donthu[2]

[1*,] Assistant Professor, Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Green fields, Guntur, Andhra Pradesh, India -522302

[2*,] Assistant Professor,School of Business,GITAM University,Bangalore, , Andhra Pradesh, India.

[1*]bvenki289@gmail.com, somuecom@gmail.com

DOI : 10.48047/IJFANS/V11/ISS10/421

Abstract:

It appears that you have presented a dataset that includes business descriptions of Brazilian enterprises that are classified into several economic activities. You wish to reduce the size of the data matrix without sacrificing any significant information by doing dimensionality reduction. This is an overview of the points you raised. Dataset Overview: 1080 documents total from your dataset contain free-text business descriptions of Brazilian enterprises. The National Classification of Economic Activities is the basis for the nine distinct categories into which these descriptions are divided (CNAE). Prepositions have been eliminated, words have been transformed into their canonical forms, and each document has been represented as a vector based on word frequency. Data Reliability: With zeros occupying 99.22% of the matrix, the dataset is extremely sparse. This indicates that a high dimensionality issue results from the majority of terms not appearing in the majority of documents. Reducing the number of variables or features in order to address the high dimensionality issue is known as dimensionality reduction. It is separated into two categories: feature extraction and feature selection. Engineering and Feature Extraction: The process of feature extraction converts unprocessed data into features that can be used in modelling. The process of increasing data correctness for algorithms is called feature transformation. In feature selection, superfluous characteristics must be eliminated. primary goal Reducing the dimensionality of the data matrix while preserving crucial information is your main objective. This entails removing features or terminology while keeping as much important data as you can. In a vector space, vector S. Tempo Model: The texts in the database are represented by you using a vector space model, in which every term becomes a dimension. Weighting Terms: By identifying terms with the highest power of discrimination and removing fewer terms, you are using term weighting approaches to increase dimensionality reduction and choose the most relevant terms.

Introduction:

instructive vocabulary. To efficiently reduce dimensionality, you may want to take into account the following methods: Term Frequency-Inverse Document Frequency, or TF-IDF: One popular method for phrase weighting is TF-IDF. It gives terms weights according on how frequently they appear in documents and how significant they are within the corpus. Terms with high TF-IDF weights are more discriminating. Latent Semantic Analysis (LSA) is a feature extraction technique that minimizes the number of dimensions in the data while maintaining semantic significance. Principal Component Analysis (PCA) is a dimensionality reduction method that facilitates the identification of the most significant words or components within the data. Algorithms for Selecting Features: To choose the most pertinent terms, apply techniques such as mutual information, chi-squared testing, or recursive feature removal. Word Embeddings: To represent words in dense vector spaces, think about utilizing

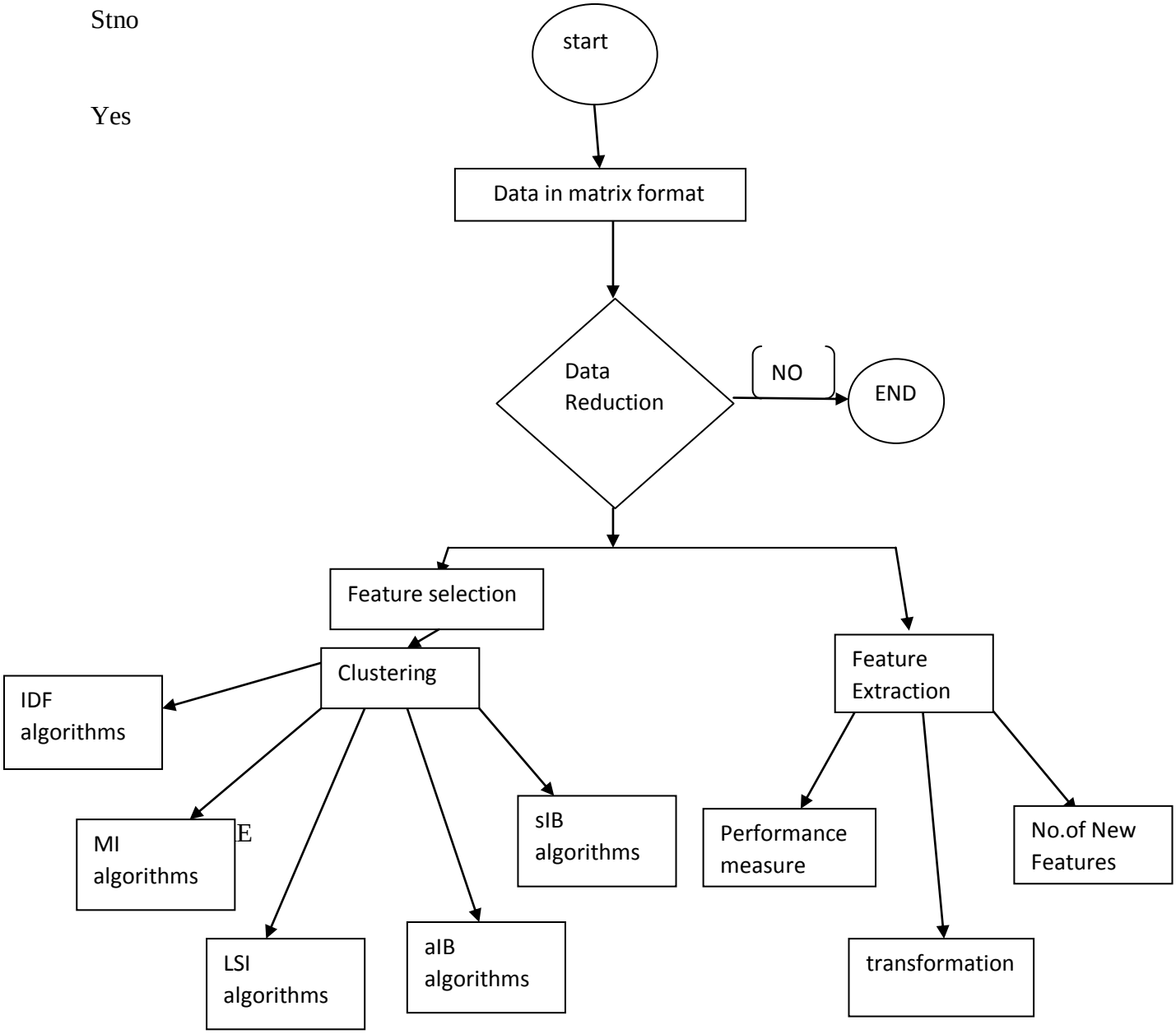
ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

word embeddings like Word2Vec or Fast Text. These embeddings can minimize dimensionality and capture semantic links. Clustering: To decrease dimensionality while preserving conceptual groups, group terms together. The type of data you have and your particular objectives will determine which dimensionality reduction strategy is best for you. It's critical to test out various approaches in order to determine which one best fits your goals and dataset. You might also need to adjust the parameters and assess how they affect your text analysis system's performance.

Proposed Model:

Stno

Yes



ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

It seems that you are talking about dimensionality reduction and word selection in relation to a vector space model that is used to represent texts in a database. Let's dissect the main ideas and methods presented:

Text data in a database is represented using the Vector Space Model (VSM). Every text is represented as a vector, and the different terms (words) that are contained in the text are represented by the vector's dimensions. Each term in the vector might have a different weight assigned to it depending on a number of parameters, including term frequency (TF), whether the term is present or absent (binary representation), or more sophisticated methods like TF-IDF.

Reducing the high dimensionality of the text data while keeping pertinent information is the aim of dimensionality reduction. Enhancing the efficacy and efficiency of text analysis jobs requires this.

Feature (Term) Selection: Selecting a subset of terms from the database that are most pertinent to the given task—text categorization, for example—is known as feature selection. The terms chosen should aid in distinguishing between various text types or categories. There are two basic approaches to feature selection: the filter approach and the wrapper approach. Filter Approach: Terms are chosen apart from the learning strategy that will employ them. This method ranks and chooses phrases based on information-theoretic or statistical measures. Wrapper Approach: A learning method that determines the ideal term combination is used to select terms. Usually, this method entails a search or optimization procedure to identify the most discriminating group of phrases.

Text Analysis Tasks: Among the most typical tasks in text analysis is clustering. It entails putting texts or phrases that are similar together. You brought up two categories of clustering:

Soft Cluster: Terms can have overlapped or hazy memberships since they can be a part of multiple groups or categories.

Hard Cluster: There is a rigid assignment since terms are linked to a single distinct group or category.

Measuring the significance of each term in the categorization of text data is the primary objective of feature selection. This helps by concentrating on the most relevant terms, which enhances performance on tasks like document classification.

Mutual Information (MI): A measure of statistical reliance or association between two variables (in this case, words), mutual information is used to quantify the relationship. There is a substantial correlation between two terms when the MI value is large. The terms that have a strong correlation with the classification or categorization of texts can be found using this metric. To minimize dimensionality, terms with high MI values are kept in place while terms with low MI values are eliminated.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

Mutual Information and other feature selection techniques can be used in practice to find and keep terms that are most informative for text classification and other natural language processing tasks. This reduces the dimensionality of the data and ultimately improves system performance. The issue of superfluous or irrelevant terms in the vector space model is addressed with the use of these strategies.

$$MI_x = p(x) \sum_{y \in ST} p(y \mid x) \log \left(\frac{p(x \mid y)}{p(y)} \right)$$

Two methods—Inverse Document Frequency (IDF) and Agglomerative Information Bottleneck (aIB) and Sequential Information Bottleneck (sIB)—as well as two clustering approaches are covered in the information you provided. These methods are applied to clustering and text data processing. Let's examine each of them individually:

Inverse Document Frequency (IDF):

The goal of the traditional term-weighting method known as IDF is to provide weights to words in a text according to how frequently those words appear in a group of documents. Giving fewer common words a higher weight throughout the entire document collection is the goal.

The formula is IDFx = log(N / nx), where nx is the number of documents that include the word x, x is the word, and N is the total number of documents in the database.

Use: It assists in locating and giving words that are uncommon or specific in the dataset a higher priority. Words that are often used in texts are given a lesser weight.

Information bottleneck aggregation (aIB):

PurposeInformation Agglomeration A clustering technique called bottleneck is used to combine clusters so as to reduce the loss of mutual information. To lower the total number of clusters, repeatedly combining clusters is started with each word (phrase) in a single cluster.

Approach: To create new clusters, the algorithm first combines pairs of clusters, trying to reduce the loss of mutual information. Until the required number of clusters is reached, this process is repeated.

Parameter: The parameter β controls the trade-off between compression and precision, as well as the softness of the resulting classification.

Goal: The goal is to reduce the number of clusters while preserving as much relevant information as possible, resulting in a more compact representation.

Sequential Information Bottleneck (sIB):

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

Purpose: Sequential Information Bottleneck is another clustering approach that aims to partition words into clusters to improve the overall performance of the partitioning.

Approach Words are first divided into a predetermined number of clusters at random. Every time, a word is transferred to create a new cluster by removing it from its existing cluster. These new clusters are then combined with preexisting clusters using a greedy approach to strengthen the division.

Multiple Solutions: Since this method can converge to local optima, the procedure is repeated multiple times (H times) to obtain different solutions. The best-performing solution is chosen as the output.

Goal: The goal is to find an optimal partitioning of words into clusters that enhances the performance of the clustering task.

Both aIB and sIB are clustering methods that use the Information Bottleneck framework, which focuses on optimizing the balance between preserving relevant information and achieving a more condensed portrayal. In order to increase the effectiveness and efficiency of clustering and classification tasks, these approaches are frequently applied in natural language processing and text data analysis. The Information Bottleneck (IB) technique is a methodology for determining the best balance between complexity (compression) and accuracy (compression) when combining a meaningful variable (Y) and a random variable (X) based on their combined probability distribution. Applications of this approach include dimension reduction and distributional clustering. The following are some essential details about the Information Bottleneck method:

Purpose: Finding a balance between maintaining pertinent information (accuracy) and simplifying a dataset (compression) is the main objective of the information bottleneck.
Joint Probability Distribution: The method works with the joint probability distribution p(X, Y), where X is the variable of interest, and Y is the observed relevant variable.

Generalization: The Information Bottleneck generalizes the concept of minimal sufficient statistics, which is a classical notion in parametric statistics. This generalization allows the method to work with arbitrary distributions, not necessarily of exponential form.

Sufficiency Condition: The sufficiency condition is relaxed in the Information Bottleneck method. Instead of requiring a variable to be fully sufficient, the method aims to capture a fraction of the mutual information with the relevant variable Y.

Rate Distortion Problem: The Information Bottleneck can be viewed as a rate distortion problem. In this interpretation, there is a distortion function that measures how well the relevant variable Y is predicted from a compressed representation T compared to its direct prediction from X.

Iterative Algorithm: To find the optimal trade-off between information preservation and compression, the Information Bottleneck method provides a general iterative algorithm. This algorithm allows for calculating the information curve from the joint distribution p(X, Y).

Compressed Variable: T is the acronym for the compressed variable in the Information Bottleneck framework. To determine the optimal representation of X in terms of T while

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

preserving a predetermined degree of information about Y, the algorithm minimizes a particular objective function or distortion measure.

Information theory, machine learning, and data compression are three areas where the information bottleneck method is most helpful. Finding the most succinct and informative data representations is helpful for a variety of data analysis and pattern recognition activities. Meaningful data summary and grouping are made possible by finding the ideal balance between information retention and compression.

$$\min_{p(t|x)} \; I(X;T) - \beta I(T;Y)$$

The Information Bottleneck framework involves optimizing the trade-off between preserving information and achieving compression. The specific optimization problem is often framed as follows:

I(X;T): The mutual information between X (the original data) and T (the compressed representation).

I(T;Y): The mutual information between T and Y (the relevant variable).

Finding the ideal value for the Lagrange multiplier (beta) within certain bounds will enable you to strike a balance between the significance of I(X;T) and I(T;Y). Stated differently, the goal is to maximize I(X;T) while maintaining I(T;Y) over a predetermined threshold or within a predetermined range. This is a typical restricted optimization problem, where you are looking for a function's local maximum or minimum under equality restrictions.

Regarding clustering:

Cluster Analysis (ClusteringIt involves organizing a collection of items (data points) into clusters based on how similar the items are to one another compared to the objects in other clusters. This is a basic exploratory data mining activity that is applied widely in many domains, such as pattern recognition and machine learning.

About the Agglomerative Information Bottleneck (AIB) algorithm:

AIB Algorithm: The AIB algorithm is a specific method for compressing discrete data through a greedy process of merging elements while minimizing the loss of mutual information with class labels. It starts with each data point as a separate component (cluster) and iteratively merges components to form new clusters.

Y-Information Decrease: During each merge, the Y-information decrease measures the reduction in mutual information between the data and the class labels. The goal is to minimize this decrease as much as possible.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

X-Information Decrease: This quantifies the reduction in mutual information between the data and itself (X) due to a merge. It's a measure of how much self-information is lost when merging components.

Greedy Procedure: The AIB algorithm takes a greedy approach, attempting to merge pairs of the current partition's components in order to make the "best possible merge" at each stage. Until the required number of clusters is reached, this process is repeated.

Through an iterative process of merging pairs of components, the AIB algorithm attempts to preserve as much information as possible while constructing clusters, exploring every potential configuration. While reducing information loss, this method works well for tasks like dimension reduction and data clustering.
You've provided a good overview of dimensionality reduction, its purpose, and some of the methods used in this field. Let's dive a bit deeper into the key points:

Dimensionality Reduction This procedure involves lowering the total number of random variables (features or attributes) that are taken into account in a dataset. It's frequently used to make data simpler, eliminate noise, and boost the effectiveness of several data analysis operations including regression and classification.
Feature Selection: Dimensionality reduction can be achieved through feature selection, where a subset of the original features is chosen for analysis. There are three common strategies for feature selection:

Filter Strategy: Involves evaluating and selecting features based on some criteria like information gain.
Wrapper Strategy: Features are selected or excluded based on their impact on model performance, often guided by accuracy measures.
Embedded Strategy: Features are added or removed during model building based on prediction errors.
Advantages of Dimensionality Reduction: It can lead to more accurate data analysis for tasks like regression and classification. By reducing the dimensionality, you may remove irrelevant or redundant information, making it easier for algorithms to extract patterns.

Methods of Dimensionality Reduction:

Principal Component Analysis (PCA): PCA is a widely used linear technique for dimensionality reduction. It aims to maximize the variance in the data when mapping it from a higher-dimensional space to a lower-dimensional space.

Linear Discriminant Analysis (LDA): LDA is another linear technique, often used in supervised classification problems. It seeks to maximize the separation between classes while reducing dimensionality.

Generalized Discriminant Analysis (GDA): GDA is an extension of LDA that allows for more flexibility in modelling data. It's useful when LDA assumptions are not met.

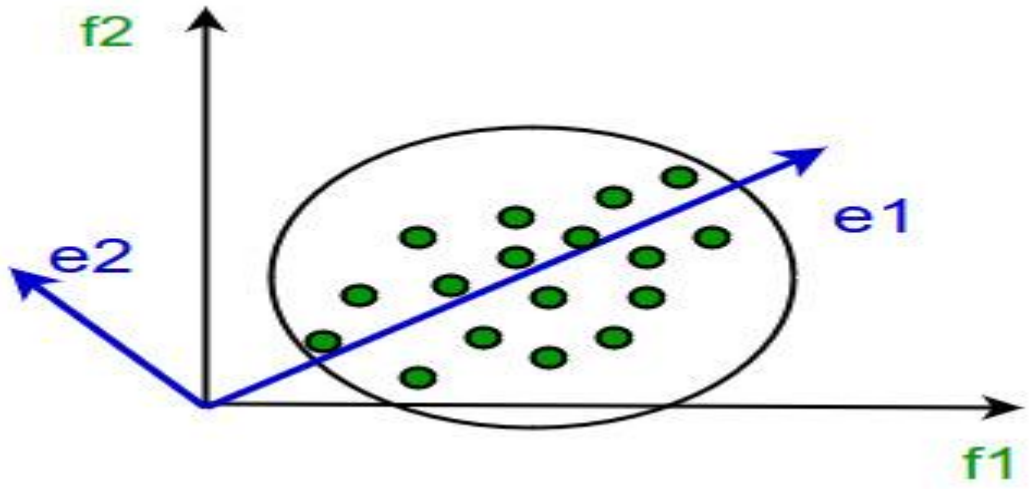
Linear vs. Non-Linear Dimensionality Reduction: Methods for reducing dimensionality can be broadly divided into two categories: linear and non-linear. One of the best examples of a

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

linear approach is PCA. In situations when there are non-linear correlations between features, non-linear techniques like Isomap and t-Distributed Stochastic Neighbour Embedding (t-SNE) are utilized.

Principal Component Analysis (PCA): Karl Pearson introduced PCA, a linear dimensionality reduction technique. It is predicated on the notion that the variance of the data in the lower-dimensional space ought to be maximized when data is mapped from a higher-dimensional space to a lower-dimensional one. Principal components, or linear combinations of features, that capture the most variance in the data are found using PCA.

All things considered, dimensionality reduction is a useful method for streamlining intricate datasets, lowering processing requirements, and enhancing machine learning model performance—especially with high-dimensional data. The particulars of the data and the analysis's objectives determine which approach is best.



You've given thorough explanations of linear discriminant analysis (LDA) and its underlying presumptions. In statistics, pattern recognition, and machine learning, linear combination analysis (LDA) is a commonly used technique for identifying feature combinations that efficiently divide data into distinct classes or groups. Let's review the main ideas:

Linear Discriminant Analysis (LDA):

LDA is a method for finding a linear combination of features that characterizes or separates two or more classes of objects or events.

It's widely used for dimensionality reduction and classification tasks.

The primary goal of LDA is to maximize the separation between classes while minimizing the spread of data within each class.

Fisher's Proposal for Separability:

LDA is based on a proposal by Ronald Fisher to maximize the function that represents the difference between the means of classes, normalized by a measure of the within-class variability.

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 10, 2022

Fisher's idea is to maximize the distance between the means of each class and minimize the spreading of data points within each class.

Assumptions of Discriminant Analysis:

Multivariate Normality: Independent variables should be normally distributed for each level of the grouping variable.

Homogeneity of Variance/Covariance (Homoscedasticity): Variances among group variables should be the same across levels of predictors. This can be tested with Box's M statistic.

Multicollinearity: The predictive power of LDA may decrease with increased correlation between predictor variables.

Independence: Participants are assumed to be randomly sampled, and a participant's score on one variable should be independent of scores on that variable for all other participants.

Robustness of Discriminant Analysis:

Discriminant analysis is relatively robust to slight violations of these assumptions.

It can still be reliable even when using dichotomous variables, where multivariate normality is often violated.

All things considered, LDA is an effective method for classification, dimensionality reduction, and feature selection. Finding the ideal linear feature combination to divide classes is the goal of this useful tool, which finds application in a number of domains such as pattern recognition, machine learning, and statistics. But, it's critical to understand the presumptions and data qualities that could have an impact on how well LDA performs in real-world scenarios.

CODE:

```
install.packages ("tidyverse")
library(tidyverse)
install.packages("MASS")
library(MASS)
install.packages("klaR")
library(klaR)
set.seed(101)
sample_n(data, 10)
setwd("C:/Users/good/Desktop")
```

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

```
data<-read.csv("vehicle1.csv", stringsAsFactor = FALSE )
training_sample <- sample(c(TRUE, FALSE), nrow(data), replace = T, prob = c(0.6,0.4))
train <- data[training_sample, ]
test <- data[!training_sample, ]
lda.data <- lda(class ~ ., train)
lda.data
plot(lda.data, col = as.integer(train$class))
plot(lda.data, dimen = 1)
partimat(class ~  X1+ X2+ X3+ X4+ X5, data=train, method="lda")
lda.train <- predict(lda.data)
train$lda <- lda.train$class
table(train$lda,train$class)
lda.test <- predict(lda.data,test)
test$lda <- lda.test$class
table(test$lda,test$class)

r <- lda(formula = class ~ .,
         data = data,
         prior = c(1, 1,1)/3)
prop = r$svd^2/sum(r$svd^2)
prop
r2 <- lda(formula = class ~ .,
          data = data,
          prior = c(1, 1,1)/3,
          CV = TRUE)
train <- sample(1:150, 75)

r3 <- lda(class ~ .,
          data,
          prior = c(1,1,1)/3,
          subset = train)
```

ISSN PRINT 2319 1775 Online 2320 7876

Research paper Volume 11, Iss 10, 2022

```
plda = predict(object = r,
               newdata = data[-train, ])
head(plda$class)
head(plda$posterior,4)
head(plda$x, 3)
```

output :

```
> table(train$lda,train$class)
```

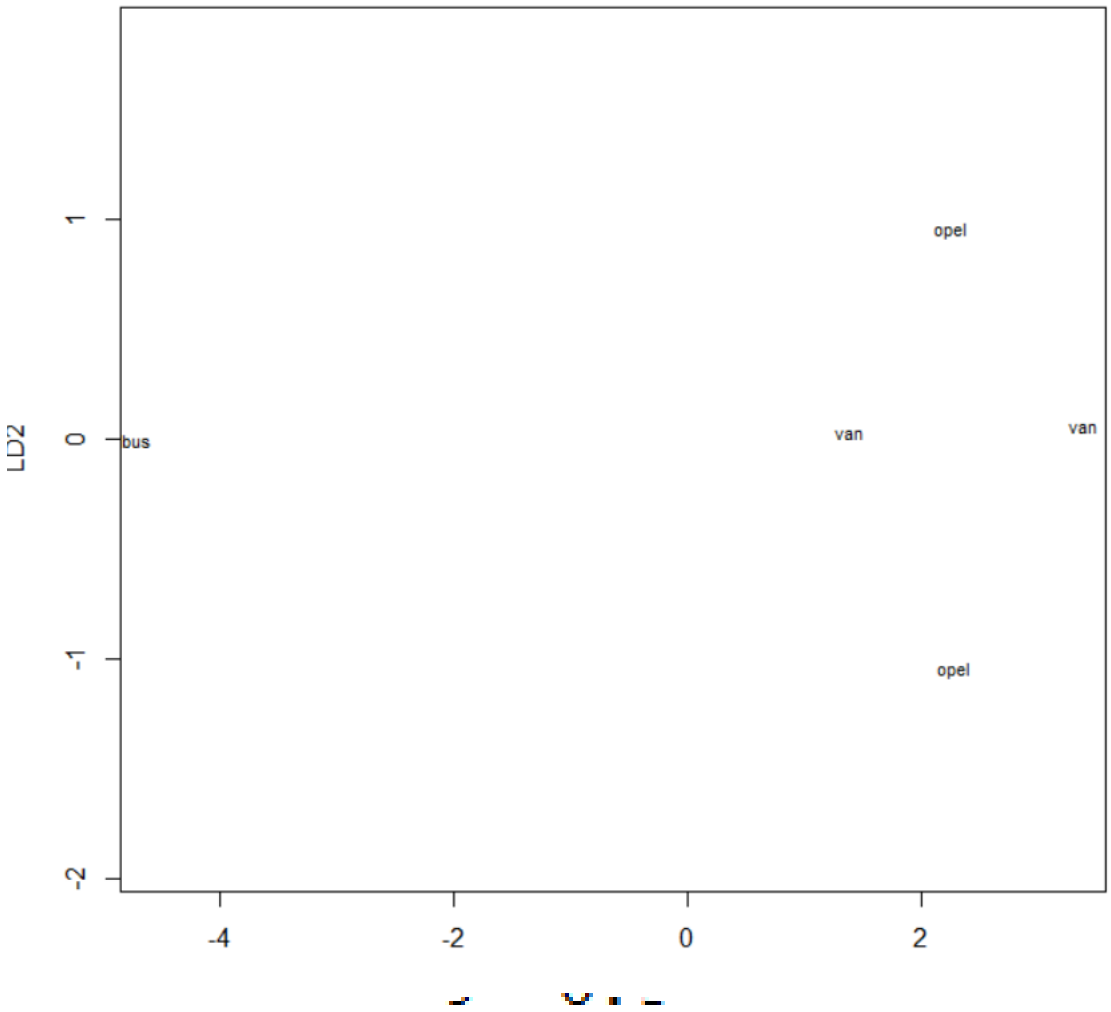
	bus	opel	van
bus	1	0	0
opel	0	1	1
van	0	1	1

Output:

```
> lda.data
Call:
lda(class ~ ., data = train)

Prior probabilities of groups:
bus  opel van
 0.2  0.4  0.4

Group means:
       X1 X2   X3 X4   X5 X6   X7 X8   X9  X10  X11  X12  X13  X14  X15  X16  X17
bus  87.0 44 65.0 44 65.0 44 65.0 44 65.0 65.0 65.0 65.0 87.0 87.0 87.0 87.0 87.0
opel 93.5 42 82.5 42 82.5 42 82.5 42 82.5 82.5 82.5 82.5 93.5 93.5 93.5 93.5 93.5
van  94.5 42 76.5 42 76.5 42 76.5 42 76.5 76.5 76.5 76.5 94.5 94.5 94.5 94.5 94.5

Coefficients of linear discriminants:
            LD1          LD2
X1   0.10660267  0.012025815
X2  -0.17345687  0.059098351
X3   0.01142973 -0.000342408
X4  -0.17345687  0.059098351
X5   0.01142973 -0.000342408
X6  -0.17345687  0.059098351
X7   0.01142973 -0.000342408
X8  -0.17345687  0.059098351
X9   0.01142973 -0.000342408
X10  0.01142973 -0.000342408
X11  0.01142973 -0.000342408
X12  0.01142973 -0.000342408
X13  0.10660267  0.012025815
X14  0.10660267  0.012025815
X15  0.10660267  0.012025815
X16  0.10660267  0.012025815
X17  0.10660267  0.012025815

Proportion of trace:
   LD1    LD2
0.9998 0.0002
>
```

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

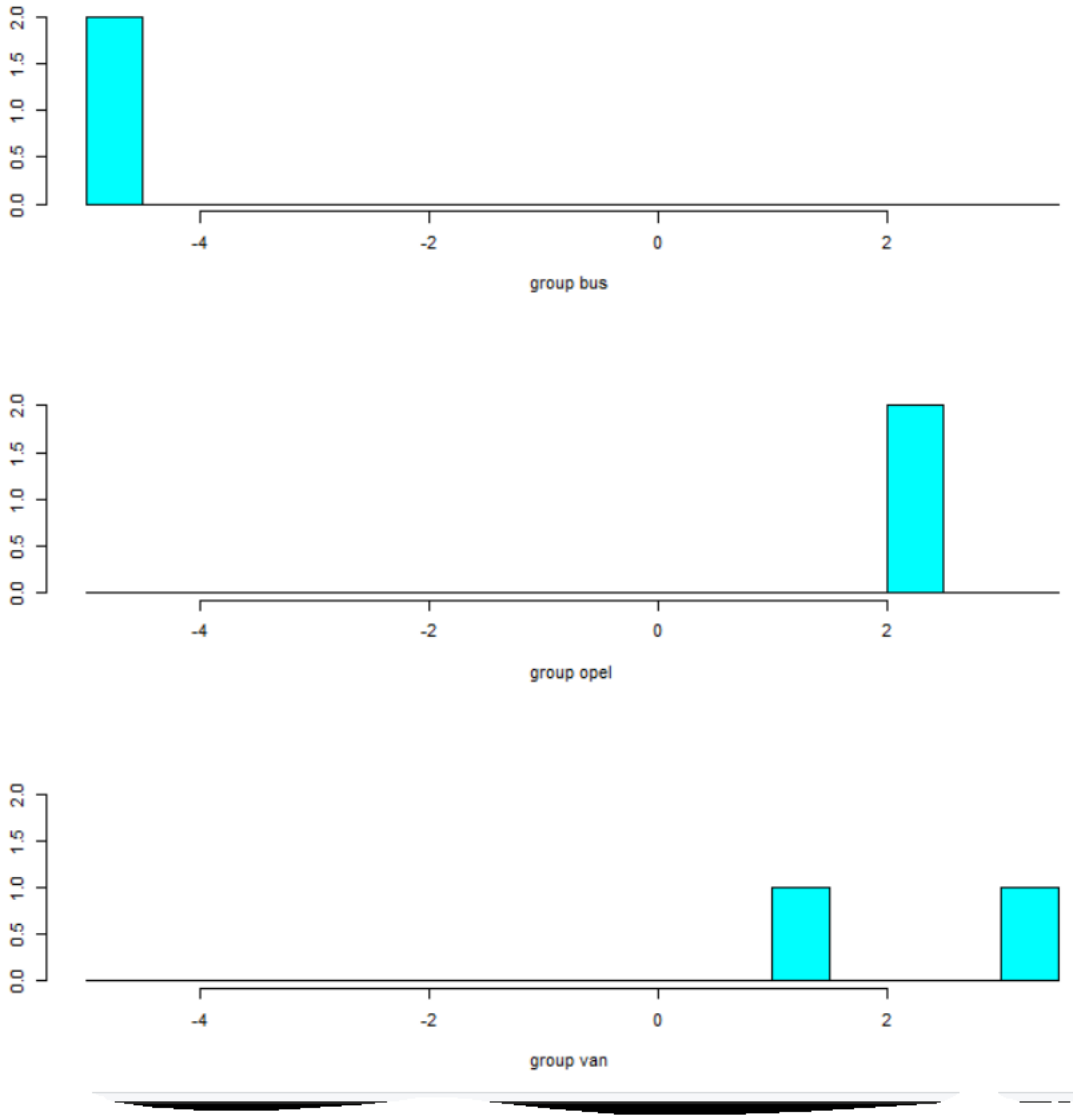
Result:




```
> prop
[1] 0.95255613 0.04744387
>
```

ISSN PRINT 2319 1775 Online 2320 7876

Research paper Volume 11, Iss 10, 2022




```
> head(plda$class)
[1] bus  bus  bus  opel van
Levels: bus  opel van
> head(plda$posterior, 3)
       bus          opel         van
4 0.9814295 0.0126706234 5.899899e-03
6 0.9889346 0.0061792649 4.886096e-03
7 0.9999115 0.0000634688 2.498478e-05
> head(plda$x, 3)
        LD1         LD2
4 -1.945120 -0.03356698
6 -2.074029  0.61917762
7 -3.559870  0.23134000
>
```

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

Conclusion and Future Scope:

To sum up, the dataset you have outlined includes company profiles of Brazilian enterprises that have been divided into various economic sectors. To deal with the significant sparsity and dimensionality problems this dataset presents, you have started a dimensionality reduction task. Text analysis and modelling can be made more successful and efficient by utilizing the dimensionality reduction strategies you described. This procedure aids in data streamlining, feature removal, and information retention.

Vector space models, term weighting, and the use of several dimensionality reduction approaches, such Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), have all been covered. These techniques are useful for turning unprocessed textual data into more readable and understandable representations.

Future Scope:

Algorithm Selection: Experiment with various dimensionality reduction techniques, including those created especially for text data, like Word Embeddings and Latent Semantic Analysis (LSA). Various approaches may provide different outcomes, so it's critical to determine which is best for your situation. dataset Hyperparameter Tuning: To maximize the effectiveness of the dimensionality reduction methods you've selected, adjust their parameters.

References:

1. W.E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. Quarterly of Applied Mathematics, 9:17–25, 1951.
2. G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis. Annotated bibliography on graph drawing. Computational Geometry: Theory and Applications, 4:235–282, 1994.
3. M. Belkin and P. Niyogi. Laplacian Eigenmaps and spectral techniques for embedding and cluster- ing. In Advances in Neural Information Processing Systems, volume 14, pages 585–591, Cam- bridge, MA, USA, 2002. The MIT Press.
4. Y. Bengio. Learning deep architectures for AI. Technical Report 1312, Universite´ de Montre´al, 2007.
5. N. Biggs. Algebraic graph theory. In Cambridge Tracts in Mathematics, volume 67. Cambridge University Press, 1974.
6. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association, 68:361–368, 1973.
7. J.A. Cook, I. Sutskever, A. Mnih, and G.E. Hinton. Visualizing similarity data with a mixture of maps. In Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, volume 2, pages 67–74, 2007.
8. M.C. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. IEEE Transactions on Visualization and Computer Graphics, 9(3):378–394, 2003.
9. V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In Advances in Neural Information Processing Systems, volume 15, pages 721–728, Cambridge, MA, USA, 2003. The MIT Press.
10. P. Demartines and J. He´rault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. IEEE Transactions on Neural Networks, 8(1):148–154, 1997.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 10, 2022

11. P. Doyle and L. Snell. Random walks and electric networks. In Carus mathematical monographs, volume 22. Mathematical Association of America, 1984.
12. D.R. Fokkema, G.L.G. Sleijpen, and H.A. van der Vorst. Jacobi–Davidson style QR and QZ algo- rithms for the reduction of matrix pencils. SIAM Journal on Scientific Computing, 20(1):94–125, 1999.
13. L. Grady. Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(11):1768–1783, 2006.
14. G.E. Hinton and S.T. Roweis. Stochastic Neighbor Embedding. In Advances in Neural Information Processing Systems, volume 15, pages 833–840, Cambridge, MA, USA, 2002. The MIT Press.
15. G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.