Discovering Product User-Experience Problem Using Unsupervised Learning

by

Jiahua You

Contents

Executive Summary

Customer support forums are a vital part of most businesses in the technology domain. They provide both service to and feedback from the customer base. They are thus an important source of data to be analyzed using knowledge engineering tools. The purpose of this project and report was to develop and utilize machine learning algorithms on data from Cisco Support Community (CSC) discussion threads, with the overall goal to automate the manual classification of computer network user-experience problems.

The first major goal of this research is to determine the best preprocessing infrastructure for unsupervised learning in the product domain of interest. In the case of our application domain of VPN products, we also have access to labeled customer data and therefore, we can apply supervised leaning techniques. First, we Identify the discreate categories in the dataset using expert domain knowledge and generate classification scheme. Second, we manually label the entire dataset using the classification scheme developed from domain knowledge. Third, we convert the data from text to vector by applying a set of data cleaning and preprocessing techniques. Fourth, we use supervised learning methods to find the best preprocessing techniques for the dataset.

To use as a data set for supervised learning analysis using logistic regression and support vector machines (SVM). We cleaned the data and used a term frequency inverse document frequency (TF-IDF) vectorizer to apply a weighting factor representing the importance of each word within each data set.

Once we developed our machine learning algorithms, we applied them to our collected data sets with a variety of implementations. For our K-Means clustering, we performed analyses on inertia and silhouette scores, and set up a decision tree classifier on the word vectors in order to assess relevance. For our supervised machine learning algorithms (logistic regression and SVM), decided to use stratified k-fold cross-validation to maximize the applicability of our supervised learning models to future, independent data sets. We implemented SVM with both polynomial and radial basis frequency (RBF) kernels. We first ran both supervised learning algorithms on our 5-category data set. We then ran them on a 2-category data set consisting of the majority class (Customer Education/Configuration Assistance) versus all other classes, in order to establish an estimated upper bound on our multinomial, 5-category analysis. Finally, we ran a pilot example of up-sampling with each supervised learning algorithm, where we increased each class size to equal the size of the majority class.

Next, we implement supervising learning techniques to find the ideal number of clusters in the dataset based on appropriate metrics. In order to find the ideal number of clusters in a dataset is a big part of the discovery process. Unless we are capable of finding the ideal number of clusters it is not possible to identify the categories of problems in the dataset. In this step we apply unsupervised learning algorithm (K-Means) and record performance using different performance metrices (Silhouette and Inertia scores) with values of clusters within a certain range (we used number of clusters ranging from 2 to 20). Comparing these recorded performance scores, we try to identify the optimal number of clusters in the dataset.

Then, we validated the output of the previous step (i.e. understanding the capability of our system to identify the number of discrete clusters). We perform visualization techniques and supervised learning approach on the same dataset aided with manual classification. This step involves the following sub-steps:

1. Label the data with new classification scheme based on the derived number of clusters from unsupervised learning technique.
2. Use data visualization techniques to explore data separability in order to find the best possible classification scheme and hence validate supervised learning results.
3. Use supervised machine learning techniques on the cleaned and preprocessed data with new classification scheme to verify unsupervised learning results.

Finally, we use of the Keywords for each identified cluster to infer the product problem associated with it. On finding and verifying the ideal number of clusters it is important to provide meaning to the derived clusters which makes the discovery complete. We use Latent Dirichlet Allocation (LDA) to form a distribution of topics within clusters and a distribution of words within each topic. Based on these distributions we find the top words (i.e. most related words) in each cluster and try to manually interpret the contents of data points within each cluster.

1 Introduction

This report addresses the development and application of unsupervised machine learning techniques to discover product user-experience problems. The purpose of this section is to properly motivate this problem, formulate the problem to be addressed, outline the research issues involved and provide a map of the research.

1.1 Background Information:

Most technology companies have customer support forums that enable customers to report their experience, and in particular, the problems encountered in using the company's products. These problems could then be used by the Technical Support team to provide solutions that help resolve the problem. A crucial first step in this process is the correct identification of the problem from the customer's unstructured (or free-form text) description of the same. This step is typically done manually by a set of subject-matter experts. Since the manual process is time-extensive, costly, an important issue is whether machine learning can be used to automate the current largely manual task of(correctly) identifying product problems from informal, unstructured customer reports (or posts) on customer support forums. Therefore, an even more important and challenging issue is whether unsupervised machine learning techniques can be used to discover product problems directly from customer reports. The successful automation of this task would result in rapid, cost-effective, and scalable solutions to customer product problems.

1.2 Problem Statement

Develop and apply a structured process for using unsupervised machine learning to discover product user-experience problems. In addition, find the capability of unsupervised learning to discover meaningful clusters.

Whether unsupervised machine learning techniques can be used to discover product problems directly from customer reports.

1.3 Research issue:

The application of unsupervised machine learning to discover customer product problems

1. How supervised learning techniques can be applied to determine the best pre-processing technique for VPN product.
2. Unsupervised learning implementation for the product domain of VPN products and other domain of interest.
3. The results of applying the Unsupervised techniques to the VPN domain including certification.
4. The results of applying the Unsupervised techniques to the VPN domain including interpretation.

2 Organization of Report

This report is organized as follows. In Section 3, we discuss the research issues and problems investigated in this project. In section 4, we introduce the knowledge engineering setup and general approach. In Section 5, we show how supervised learning techniques can be applied to determine the best pre-processing technique for VPN product. In section 6, we describe unsupervised learning implementation for the product domain of VPN products and other domain of interest. And, the results of applying the Unsupervised techniques to the VPN domain including certification. The results of applying the Unsupervised techniques to the VPN domain including interpretation. Conclusions and suggestions for next step are describe in Section 7. Appendix is in Section 8.

3 Research Issues

3.1 Problem Statement

Develop and apply a structured process for using unsupervised machine learning to discover product user-experience problems. In addition, find the capability of unsupervised learning to discover meaningful clusters.

Whether unsupervised machine learning techniques can be used to discover product problems directly from customer reports.

3.2 Research Issues

The application of unsupervised machine learning to discover customer product problems

1. How supervised learning techniques can be applied to determine the best pre-processing technique for VPN product.
2. Unsupervised learning implementation for the product domain of VPN products and other domain of interest.
3. The results of applying the Unsupervised techniques to the VPN domain including certification.
4. The results of applying the Unsupervised techniques to the VPN domain including interpretation.

4 Knowledge Engineering

In this section, we outline our knowledge engineering setup, approaches, and implementation. We include details on machine learning algorithm performance.

4.1 Knowledge Engineering Setup

Here we describe how we cleaned and represented the textual data from the Cisco Support Community (CSC) and which machine learning algorithms we used to develop our models.

4.1.1 Representing Textual Data

Of the two types of machine learning classifiers, statistical and semantic, we decided to implement the former. The team applied the Technology Domain knowledge to create a relevant vocabulary (bag-of-words) to describe VPN problems for the purpose of obtaining accurate machine learning results. The first step in representing the data was to perform word cleaning on the Cisco Support Community (CSC) data set to optimize the performance of the TF-IDF vectorizer, which performs a statistical frequency analysis of words (features).

4.2 General Approach

Our initial goal was to cut down on the noise in our data, and thereby improve results. We therefore designed and implemented a pre-processing regime to render our data more useful to our knowledge engineering algorithms. This process included cleaning and representation of our data, followed by the application of machine learning models.

4.2.1 Cleaning and Representing Textual Data

We cleaned the text data from the original post in the CSC forum, stemmed the data, then passed it through a TF-IDF vectorizer.

4.2.2 Machine Learning Models

The labeled, TF-IDF vectorized matrix representation of will be fed into our machine learning algorithms: K-means clustering, logistic regression, and support vector machines.

4.3 K-Means Clustering Approach

We wished to discern if K-Means clustering is useful, given the nature of the data and the set of VPN problems. If it is determined that clustering can be useful, the team will evaluate how it will be deployed. We hope to map one or more clusters to one or more user-experience problem categories.

4.4 K-Means Clustering Implementation

Our implementation includes analyzing the inertia and silhouette scores of different numbers of clusters, visually representing the clusters in 2 dimensions, and using a decision tree classifier to determine the relevance of our clustering results.

4.5 Supervised Machine Learning Approach

The goal of our supervised machine learning algorithms, support vector ma- chines and logistic regression, is to accurately classify CSC posts. We pass our labeled, pre-processed data through our machine learning algorithms in an attempt to create an accurate model that can be used to classify further posts. With a sufficiently accurate model, we could classify posts automatically in order to have them routed to the appropriate domain experts.

4.5.1 Problem Categories

We classified our data into our 5 problem categories:

1. VPN Tunnel Not Coming Up
2. VPN Tunnel Flapping
3. Applications Not Working Across VPN Tunnel
4. Customer Education/Configuration Assistance
5. VPN Feature Not Working

4.6 Supervised Machine Learning Implementation

In this section, we describe the specific implementations we ran for our supervised machine learning algorithms.

We classified our data into our 5 problem categories, and we ran the following supervised machine learning algorithms with 5 categories: Logistic Regression and Support Vector Machines.

We also binary classifications of our majority class (Customer Education/Configuration Assistance) vs. everything else, to find an approximate upper bound for our 5-category multinomial classifications. Finally, we attempted unsampled 5category classifications, to mitigate the imbalance of our dataset.

4.6.1 Problem Categories

For our 5-category supervised machine learning classifications, we used the following categories,

1. VPN Tunnel Not Coming Up
2. VPN Tunnel Flapping
3. Applications Not Working Across VPN Tunnel
4. Customer Education/Configuration Assistance
5. VPN Feature Not Working

4.6.2 Dataset

The dataset is a collection of 668 posts from the Cisco Support Community (CSC), which are successfully answered and selected randomly from the most recent 3 years (2015- 2017) of cases. It consists of unstructured text data within the Virtual Private Network domain.

The discrete 5 categories (mentioned in the earlier section) accurately reflect the types of problems and challenges users experience with Cisco VPN products.

4.6.3 Data Analysis and Category Schemes

The categories as described in the previous subsection were used to label the 668 posts from the Cisco Support Community.

The labels are 1 through 5 denoting the categories mentioned in “Category Identification”. On finding the data distribution according to the “5CAT” classification scheme, we noticed that there was a large imbalance in the dataset as well as more than one small imbalance in the dataset. Category 4 was the dominating category with 446 posts and the remaining 4 categories formed other 222 posts.

Post Count and Frequency by n-value		
Category	5CAT	2CAT
1	56	222
2	16	446
3	107	-
4	446	-
5	43	-

Table I
DATA DISTRIBUTION WITH CATEGORY

The Table 1 shows the category distribution of “5CAT” classification scheme. On further analysis of the distribution we found that category 4 (the dominant category) is approximately 66 percent of the entire dataset. The largest category of the 4 other categories (categories 1,2,3 and 5) has 107 posts which is around 23 percent of the dominant category and 16 percent of the entire dataset.

The main idea behind this step is that even though it is possible to split the data into 5 discrete categories, the machine learning algorithms are only able to detect 2 discrete categories.

We have enough supporting evidence of this classification scheme from the results of data visualization (described in the results section) and results of the supervised learning section with ”2CAT” classification scheme (which will be described in the results section of supervised learning). The two categories are labeled as the following:

- Customer Education issues
- Customer problem with connection

The Table 1 “2CAT” shows us the category distribution after recategorization of the entire dataset.

The 1st category “Customer Education Problems” has 222 posts and “Customer Problems with Connection” has 446 posts.

The second category is directly derived from category 4 (the dominant category) from the “5CAT” classification scheme.

The posts from remaining 4 categories from the ”5CAT” classification scheme (categories 1,2,3 and 5) were combined to form the first category.

The merge was done in this fashion since categories 1,2,3 and 5 in the “5CAT” classification scheme are subcategories of “Customer Education Problems”.

5 Supervised Learning

5.1 Supervise Learning Approach & Best Pre-processing Step

The primary goal and purpose of supervised learning in this research is determine the best pre-processing technique in order to run unsupervised learning algorithms on preprocessed data. Hence, we need to,

- Identify the discreate categories in the dataset using expert domain knowledge and generate classification scheme.
- Manually label the entire dataset using the classification scheme developed from domain knowledge.
- Convert data from text to vectors by applying a set of data cleaning and preprocessing techniques.

We performed cross validation by splitting up the train data into training and validation sets. On finding the best pre-processing technique using validation set we verified the same by training the entire train set and testing it on a separate test set. The entire process is discussed in detail in the following subsections.

5.1.1 Train test split

The train data was used for cross validation. The test data was used after training the entire train data with hyper parameters achieved from the validation set.

Note, in this case, the test set is to validate the results further and not just used measure performance.

Pointer → This is explained in supervised learning results subsection.

5.1.2 Stratified 10 Folds Cross Validation

Since the data is slightly imbalanced even after applying data imbalance techniques, one of the most crucial techniques is to use, stratified k folds cross validation.

In this way we could guarantee that every fold is a good representative of the entire dataset. This was performed over the train dataset.

The purpose was to find a **regularized** and **unbiased** result of the trained weights which could be used to find the test accuracy from the test set.

5.1.3 Learning Algorithms (Supervised Learning)

For selecting learning algorithms, we chose different genres of classifiers sequentially adding more complexities to handle non-linearity of the data. This could provide us more insight of the capability of the learning algorithm to handle the complexity of the data.

The chosen algorithms have also been tried out with different hyper parameters to find out even a nuanced version of the training algorithm. The training algorithms chosen are as follows:

- Logistic Regression
- Random Forests
- KNN (K nearest Neighbors)
- SVM (Support Vector Machines)

5.1.4 Training, testing and recording

Once the best pre-processing method and learning algorithm was found the entire training data set was used to train the model and update the weights using the best classifier.

On training the data set we tried recording the training error as well making sure that there is no hint of data overfitting.

The next step was to test the trained model on the test dataset to verify our results from cross validation.

5.2 Supervised Learning Results

Even though our primary goal of supervised learning results was to find the best pre-processing technique, as a bi-product of this step, we get to show that the 2CAT classification scheme produces better results among the classifiers compared to that of 5CAT classification scheme.

5.2.1 5 category schemes

The following tables represents the results, which is obtained from supervised learning. Table shows the results using 5 category scheme labeled data. The cell with the **highest accuracy** for a pre-processing and algorithm combination is highlighted in the table.

Category 5 Results			
Classifiers	tfidf	doc2vec	w-doc2vec
svm_poly	69.05	72.69	74.08
svm_linear	69.43	72.45	73.45
knn_30	68.67	72.26	72.26
knn_25	69.43	71.13	71.69
knn_20	68.30	72.45	72.26
lr_liblinear	68.86	70.56	71.13
lr_lbfgs	68.86	70.94	70.94
rf_45	68.30	71.13	72.56
rf_30	68.87	72.78	70.75
rf_20	68.31	71.77	71.64

Table 2

- KNN with k = 25 has highest accuracy of 69.43 percent for TF-IDF preprocessing,
- SVM with polynomial kernel has highest accuracy of 72.69 percent for doc2vec preprocessing,
- SVM with polynomial kernel has highest accuracy of 74.08 percent for weighted doc2vec pre-processing.

To find the best preprocessing technique for this classification scheme, we took an average over all the classifiers and compared the results. We have,

- x percent average accuracy for TF-IDF,
- y percent average accuracy for doc2vec
- z percent average accuracy for weighted doc2vec

Hence, we can say for the "5CAT" classification scheme **weighted doc2vec** is the best preprocessing technique.

5.2.2 2 category schemes

The following tables represents the results, which is obtained from supervised learning. Table shows the results using 5 category scheme labeled data. The cell with the **highest accuracy** for a pre-processing and algorithm combination is highlighted in the table.

Category 2 Results			
Classifiers	tfidf	doc2vec	w-doc2vec
svm_poly	71.27	78.88	84.62
svm_linear	71.23	79.32	83.73
knn_30	71.13	77.94	80.58
knn_25	72.09	76.26	81.29
knn_20	70.82	76.03	80.22
lr_liblinear	69.81	77.90	82.35
lr_lbfgs	70.55	77.56	81.94
rf_45	71.02	75.85	79.62
rf_30	69.84	75.78	81.84
rf_20	69.93	76.97	80.67

Table 3

From table (2 category scheme),

- KNN with k = 25 has highest accuracy of 72.09 percent for TF-IDF preprocessing,

- SVM with linear kernel has highest accuracy of 79.32 percent for doc2vec preprocessing,
- SVM with polynomial kernel has highest accuracy of 84.62 percent for weighted doc2vec preprocessing.

In order find the best preprocessing technique for this classification scheme we took an average over all the classifiers and compared the results. We have,

- x percent average accuracy for TF-IDF,
- y percent average accuracy for doc2vec
- z percent average accuracy for weighted doc2vec

Hence, we can say that even for the "2CAT" classification scheme **weighted doc2vec** is the best preprocessing technique.

5.2.3 Conclusion

Hence, as a summary of supervised learning results we have for both Category schemes Weighted Doc2Vec as the best pre-processing and SVM with polynomial kernel as the classifier for training on the entire dataset.

5.3 Summary

We verified our results on the entire training set to observe any hint of existing bias or variance in the trained algorithm. The summary of supervised learning in terms of the best results on train, test and validation set for both cases are described in the table (Supervised Learning Result).

Supervised Learning Results			
Categories	Training	Validation	Test
Category 5	75.66	74.08	73.85
Category 2	83.93	84.62	84.02

Table 4

From Table (Supervised Learning Result), we find that all three train, validation and test accuracies are **fairly close** for each classification scheme, which suggests that the trained model is free from bias or variance.

This confirms **weighted doc2vec** as the best preprocessing technique. Also, as a bi-product of this experiment we find that **"2CAT"** classification scheme provides us a 10 percent increase in accuracy when compared to "5CAT" classification scheme.

Pointer → This result is later used to verify the results from Unsupervised learning section.

6 Unsupervised Learning

6.1 Unsupervised Learning Approach

In this section, we share the results of Unsupervised Learning process, approach and final results. The same dataset (VPN) with same preprocessing techniques (TF-IDF, Doc2Vec, weighted Doc2Vec) were used for the Unsupervised Learning approach. All three pre-processed vectors were fed through learning algorithms without their assigned labels.

In order to determine the ideal number of clusters, we examine the silhouette and inertia score.

1. **Silhouette scores**, the number of clusters were detected via Silhouette scores. It can reflect the data is grouped that object are organized into groups that match it. This is a tool to assess the validity of the clustering to be used for selecting the optimal number in the cluster. Which silhouette score is used to support the evaluation clustering with the maximum of silhouette. This will help us to determine the ideal number of clusters for unsupervised learning.

2. **Inertia scores**, the number of clusters were detected via Inertia scores. It is a measure of variation, or “spread”. When we plot the explained inertia (intragroup sum of squares / total sum of squares) versus the number of clusters you usually notice an "elbow", a big curvature (or a peak in the second derivative) at some point. This is called the elbow criterion and it helps determine when it is no longer necessary to add more clusters. However, there may be many peaks and so this requires critical analysis. This will help us to determine the ideal number of clusters for unsupervised learning.

3. **LDA, Latent Dirichlet Allocation**, was used to find meaning of the clusters in terms of the most important words present in a certain cluster. A prior value of number of clusters with highest scores for K-means algorithm was provided to the LDA algorithm to get best results. Also, the LDA model will be used to find the internal information of the 2 clusters, meaning the words within the clusters will be discovered and interpreted.

We used the entire dataset in these approaches, since the goal was not to build the accuracy of the classifier and rather find the optimal number of clusters. Hence, there was no train test split step involved for the unsupervised learning approach.

6.2 Unsupervised Learning Results (K-Means)

The following figures shows us the results of K-Means. We apply unsupervised learning algorithm, K-Means, and record performance using different performance metrices

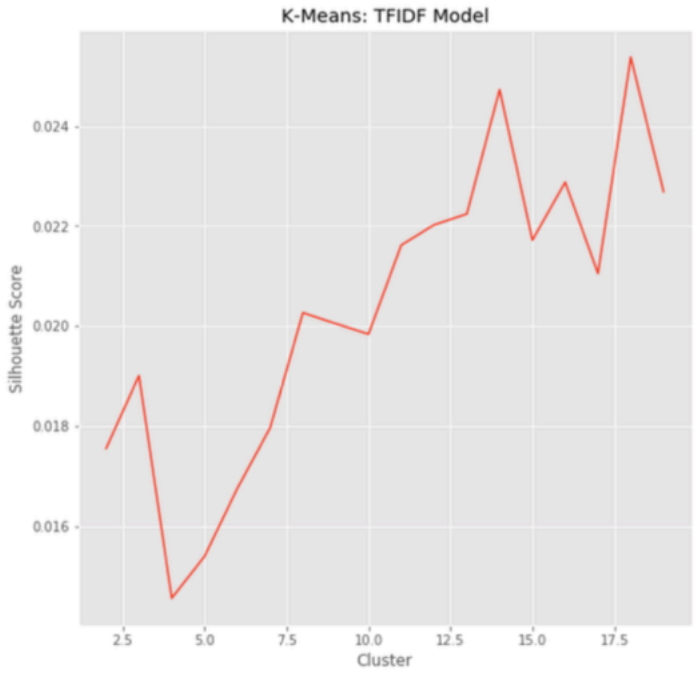
- Silhouette and comparison.
- Inertia scores comparison.

with values of clusters within a certain range (we used number of clusters ranging from 2 to 20). Comparing these recorded performance scores, we try to identify the **optimal** number of clusters in the dataset.

6.2.1 Silhouette Score Comparison

Figures a, b, c shows us the Silhouette scores for TF-IDF, Doc2vec and Weighted Doc2vec representations respectively.

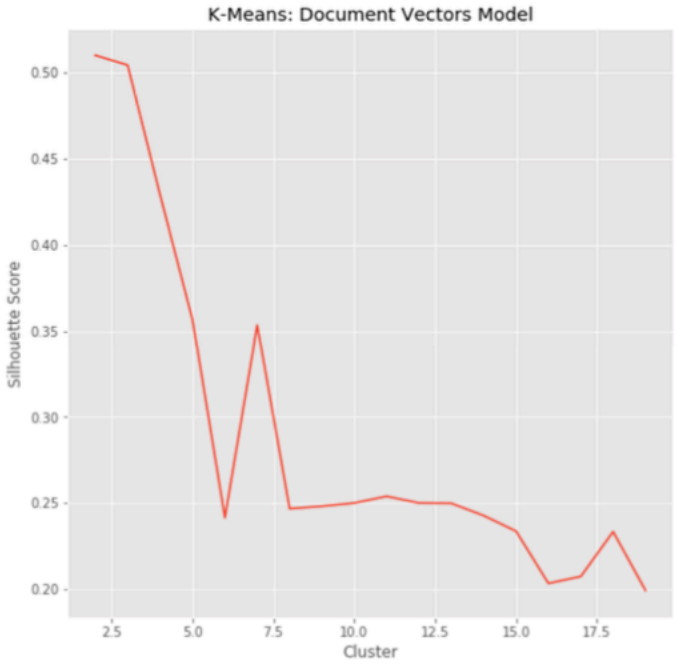
a. Silhouette score vs Cluster (TF-IDF)



The TF-IDF model shows that there is not much **cohesion** and **separation** between clusters. It points towards a few possible clusters, 3 clusters, 8 clusters, 14 clusters, 16 clusters, and 18 clusters. Since the model does **not clearly** show a perfect spot of cohesion and separation of clusters and the silhouette scores at these points are **extremely low**, it is probably not a good way to represent the data.

Cohesion refers to the *degree to which the elements inside a module belong together*. In one sense, it is a measure of the strength of relationship between the methods and data of a class and some unifying purpose or concept served by that class. In another sense, it is a measure of the strength of relationship between the class's methods and data themselves.

b. Silhouette score vs Clusters (Doc2vec)

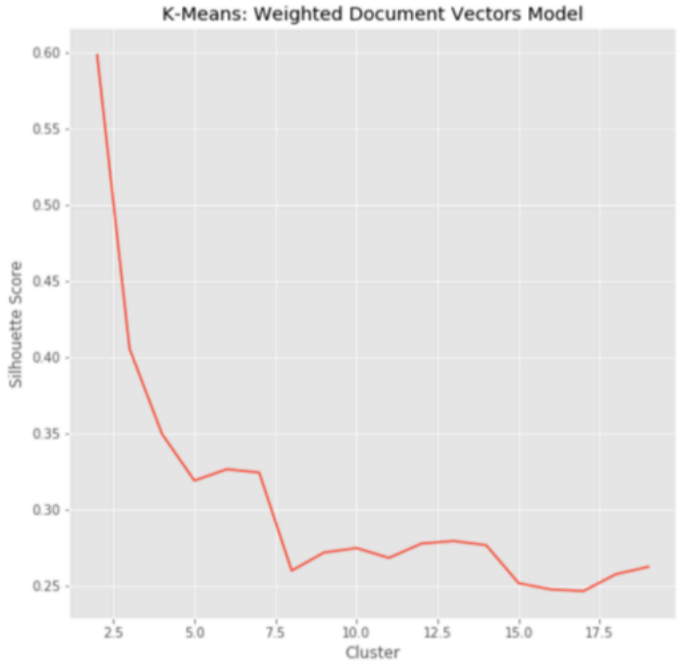


The Doc2vec, or document vectors model, shows **some cohesion** and **separation** within 2 clusters and 7 clusters. Because the model follows a **downward trajectory** and the silhouette score is highest at 2 clusters, the model suggests that 2 clusters fits the data best.

On the other hand, it is still possible that 2 clusters are not a useful number of clusters because the silhouette score is relatively low.

c. Silhouette score vs Cluster (Weighted Doc2vec)

The weighted doc2vec model seems to be the **best** model. It has the highest silhouette score, at 0.60, presenting the idea that 2 cluster may allow for distinct problem types to form.



The weighted doc2vec, or weighted document vectors model, shows some **slight cohesion** and **separation** at 2 and 7 clusters.

The silhouette score for this model is highest at 2 clusters, which found that the density of the k values k = 2 show the cohesion (density) and separation us optimal. This indicating that there may be 2 distinct problem types.

The silhouette score is **slightly high** at two clusters, compare to TF-IDF model and doc2vec model, offering the idea that this model may provide significant information.

d. Conclusion (Silhouette Score Comparison):

The weighted document vectors model seems to be the best model. It has the **highest** silhouette score, at 0.60, presenting the idea that 2 cluster may allow for distinct problem types to form.

The TF-IDF model shows the lowest silhouette score, at about 0.03 at its highest point, 18 clusters. Therefore, it seems that the TF-IDF model is not a useful way to model the data.

The doc2vec model has a high silhouette score, 0.55, at 2 clusters. However, it is still slightly lower than the silhouette score for the weighted doc2vec.

In addition, the slope of the silhouette score of the weighted doc2vec model compared to the doc2vec model, is more **steeper** from 2 to 3 clusters indicating that there is more **drastic change** in silhouette score from the two cluster groupings.

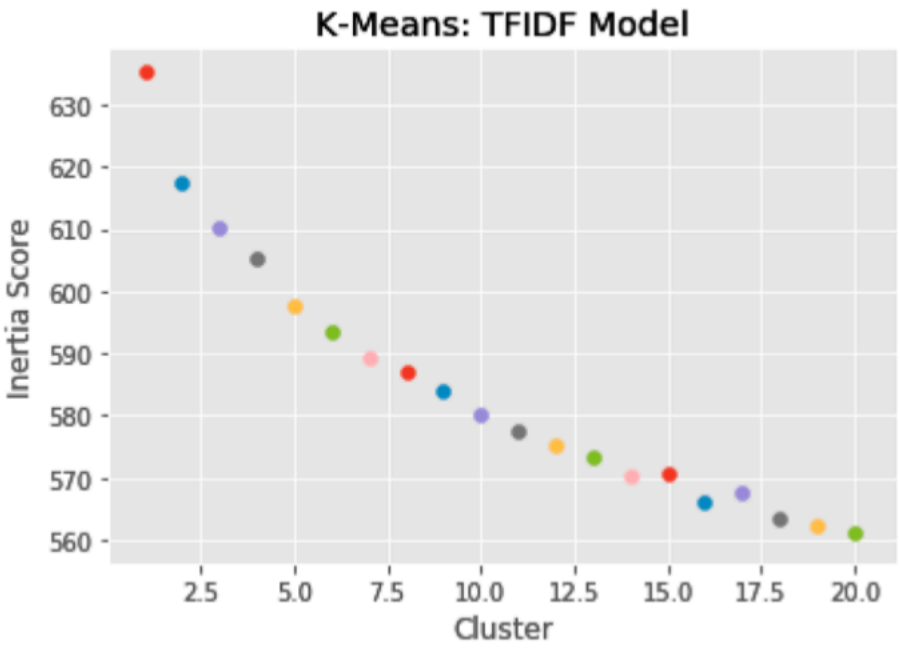
Significantly, the drastic change is important, because it proves that there is a bigger difference in **cohesion** and **separation** from 2 to 3 clusters.

Therefore, the weighted document vectors model seems to give more distinct problem types than the document vector model.

6.2.2 Inertia Score Comparison

- Figures a, b, c shows us Inertia scores for TF-IDF, Doc2vec and Weighted Doc2vec representations respectively.

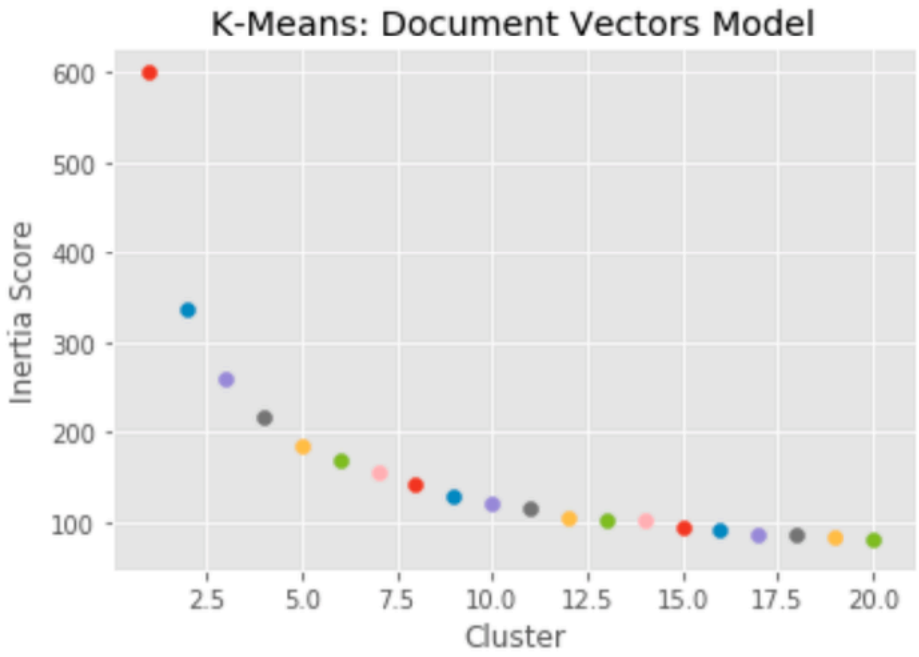
a. Clusters vs Inertia score TF-IDF



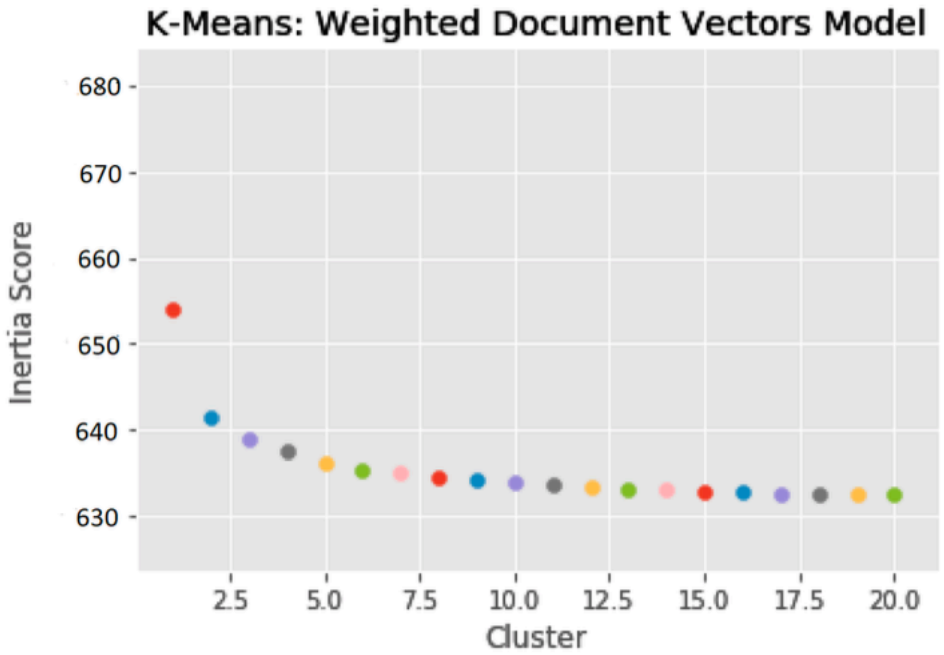
The inertia score for the TF-IDF Model, shows that the model **does not** form **cohesive** clusters because there is no "elbow," or where the variance **reduces significantly**, identified in the visualization.

Therefore, it does not show any significant groupings of clusters and should not be used as a way to find distinct problem types.

b. Clusters vs Inertia score Word2Vec



c. Clusters vs Inertia score Weighted Word2Vec



The inertia score for the **doc2vec** and the **weighted doc2vec** models show a clear indication that **cohesion** exists at 2 clusters.

The difference is that, the doc2vec model has a steeper slope of cohesion from cluster 1 to 2 and 2 to 3 and has an overall higher inertia score than the weighted doc2vec model.

d. Conclusion (Inertia Score Comparison):

The inertia score analysis shows that **both** the doc2vec model and the weighted doc2vec model may be good models for detecting distinct problem types because they both show distinctively that the variance reduces significantly, creating an “elbow,” at 2 clusters.

In addition, the **doc2vec** model may be slightly better due to the steeper slope between 2 clusters and its neighboring clusters, suggesting a greater difference in cohesion.

6.3 Summary:

The summary of unsupervised learning is as follows in terms of best Inertia and Silhouette scores with number of clusters.

Unsupervised Learning Results			
Scores	TfIdf	Doc2Vec	WDoc2Vec
Silhouette	0.024 (17 clusters)	0.52(2 clusters)	0.6(2 clusters)
Inertia	636 (2 clusters)	600(2 clusters)	650(2 clusters)

Table 5

The results show that the clusters offer more **cohesion** and **separation** in the doc2vec model and weighted doc2vec model than the TF-IDF model. This shows that representing the data differently as a pre-processing step generates better results.

Comparing results from Table, the best model seems to be the weighted doc2vec model is so far, the best model to use with this type of data. This is probably because the model uses the meaning of each word to place the document in space, allowing for model to interpret the data better. **It can also be seen that the best number of clusters is 2 clusters**.

The silhouette score is high, 0.53/0.60, and the “elbow” forms for both the document vectors model and the weighted doc2vec model.

Also, the LDA model will be used to find the internal information of the 2 clusters, meaning the words within the clusters will be discovered and interpreted.

7 Conclusion

In this section, we outline the research conclusion, supervised learning conclusion, unsupervised learning conclusion, comparison between supervised learning and unsupervised leaning, analysis of clusters, and overall summary of conclusion for each of the 4 research issues.

7.1 Research Conclusion

This research discusses the problem of product user-experience topic extraction, and it explores three ways of data preprocessing (TF-IDF, doc2vec, weighted doc2vec) and use supervised learning and unsupervised learning (K-Means) to decide the best data preprocessing method and number of classes for Cisco user-experience questions in the forum. The result shows that weighted doc2vec preprocessing method delivers best result with 2 categories.

7.1.1 Supervised Learning

We verified the results on the entire training set to observe any hint of existing bias or variance in the trained algorithm. (Bias is how far are the predicted values from the actual values. Variance tells us how scattered are the predicted value from the actual value are.) The summary of supervised learning in terms of the best results on train, test and validation set for both categories cases are described in this table (Supervised Learning Result).

Supervised Learning Results			
Categories	Training	Validation	Test
Category 5	75.66	74.08	73.85
Category 2	83.93	84.62	84.02

Table 4, Accuracy of train, validation and test result

Training dataset is the sample of data used to fit the model. ***Validation dataset*** is the sample data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on validation dataset is incorporated into the model configuration. ***Test dataset*** is the sample of data used to provide an unbiased evaluation of final model fit in the training dataset. All these number are best result from the training, validation and test, and measure the accuracy of percentage.

From Table 4, we find that all three dataset (train, validation and test) accuracies are **fairly close** for each of 2 classification schemes, which indicates that the trained model is free from bias or variance. In validation dataset we find the 74.08 is highest accuracy from category 5 results (Section 5.2.1, Table 2), and 84.62 is highest accuracy from category 2 results (Section 5.2.2, Table 3). Compare to TF-IDF, doc2vec and weighted dic2vec, we confirm **weighted doc2vec** is the best preprocessing technique across all classifiers.

In addition, as a bi-product of this experiment we find that "2CAT" classification scheme provides us a 10 percent increase in accuracy when compared to "5CAT" classification scheme (Section 5.3, Table 4). This comparison confirms supervised learning method delivers best result with 2 dominant categories for the product.

The result from supervised learning (weighted doc2vec is the best preprocessing technique, and best result with 2 dominant categories) is later used to validate and understand the unsupervised learning results.

7.1.2 Unsupervised Learning

The summary of unsupervised learning is as follows in terms of best Inertia and Silhouette scores with number of clusters.

Unsupervised Learning Results			
Scores	TfIdf	Doc2Vec	WDoc2Vec
Silhouette	0.024 (17 clusters)	0.52(2 clusters)	0.6(2 clusters)
Inertia	636 (2 clusters)	600(2 clusters)	650(2 clusters)

The results show that the clusters offer more **cohesion** and **separation** in the doc2vec model and weighted doc2vec model than the TF-IDF model. This shows that representing the data differently as a pre-processing step generates better results.

Comparing results from Table, the best model seems to be the weighted doc2vec model is so far, the best model to use with this type of data. This is probably because the model uses the meaning of each word to place the document in space, allowing for model to interpret the data better. The silhouette score is highest at 0.60, and the "elbow" forms the weighted doc2vec model.

It can also be seen that the best number of clusters is 2 clusters. This is based on using Silhouette scores and Inertia scores for the K-Means unsupervised machine learning algorithm identified 2 dominant clusters. This is based on silhouette score in Weighted Doc2vec model is highest at 0.60, at 2 clusters, presenting the idea that 2 cluster will allow for distinct problem types to form. Inertia Score for Weighted Doc2vec model show a clear indication that cohesion exist at 2 clusters, at 650. (Section 6.3, Table 5).

7.1.3 Comparison Supervised Learning and Unsupervised Learning

The results of the supervised and unsupervised learning methods match significantly. From the summaries of Supervised Learning results (Table 3) we have the 2 category scheme performing with 10 percent more accuracy (Table 4) on test set whereas the Silhouette and inertia scores from unsupervised learning suggests 2 clusters to be the best within a range from 2 to 20 clusters (Table 5). Also, in terms of preprocessing techniques, weighted Doc2vec produced the best results for both Supervised and Unsupervised learning methods.

7.4 Overall Summary

In this section we draw conclusions for each of the 4 research issues enumerated in Section 3.

1. Determination of the best pre-processing infrastructure for supervised learning in the product of interest.

   First, in this particular case, we have access to a labelled dataset, and then we were able to apply supervised learning techniques to determine that **Weighted Doc2vec** is the best preprocessing technique across all classifiers. This is based on the result of supervised learning, SVM with polynomial kernel has highest accuracy of 74.08 percent for Weighted Doc2vec preprocessing from “5CAT” result (Section 5.2.1, Table 2), SVM with polynomial kernel has highest accuracy of 84.62 percent for Weighted Doc2vec preprocessing from “2CAT” result (Section 5.2.2, Table 3).

Second, supervised learning method delivers best result with 2 dominant categories for the product. This is based on we find that “2CAT” classification scheme provides us a 10 percent increase in accuracy when compared to “5CAT” classification scheme (Section 5.3, Table 4).

2. Determination of the best pre-processing infrastructure for unsupervised learning in the product of interest.

   We compare to 3 pre-processing techniques using Silhouette scores and Inertia scores for the K-Means unsupervised machine learning algorithm. We determine that **Weighted Doc2vec** provides best preprocessing technique. This is based on silhouette score in Weighted Doc2vec model is highest at 0.60 (Section 6.3, Table 5). Inertia Score for Weighted Doc2vec model is highest at 650 (Section 6.3, Table 5).

3. Determination of the optimal number of clusters based on the appropriate performance metrics.

   Using Silhouette scores and Inertia scores for the K-Means unsupervised machine learning algorithm identified 2 dominant clusters. This is based on silhouette score in Weighted Doc2vec model is highest at 0.60, at 2 clusters, presenting the idea that 2 cluster will allow for distinct problem types to form. Inertia Score for Weighted Doc2vec model show a clear indication that cohesion exist at 2 clusters, at 650. (Section 6.3, Table 5). This result matches the 2 dominant clusters from the “2CAT” manual recategorization, Customer Education Issues and Customer

Problems with Connection (Section 4, Table 1). This match provides initial support for the capability of unsupervised learning to discover the dominant clusters.

REFERENCES

[1] Munger, Tyler, Subhas Desa, Chris Wong. (2015). The Use of Domain Knowledge Models for Effective Data Mining of Unstructured Customer Service Data in Engineering Applications. *BigDataService*. 2015: 427-438.

[2] S Laney, O Ahmed, R Glenn S Desa, (2017). Machine Learning in the Network Troubleshooting Domain. *Project Report*. Fall 2017. Baskin School of Engineering, UC Santa Cruz.

[3] Zhang S., Li A., Zhu H., Sun Q., Wang M., Zhang Y. (2018). Research on the Protocols of VPN. In *Advances in Intelligent Systems and Interactive Applications.* IISA 2017. Xhafa F., Patnaik S., Zomaya A. (Eds). Advances in Intelligent Systems and Computing, vol 686. Springer, Cham.

[4] Cisco Support Community Forums (VPN section): https://supportforums.cisco.com/t5/vpn/bd-p/6001-discussions-vpn

[5] Bird, Steven, Edward Loper, and Ewan Klein. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc.

[6] Dan Garrette and Ewan Klein. (2009). An extensible toolkit for computational semantics. In *Proceedings of the Eighth In- ternational Conference on Computational Semantics*. (IWCS-8 '09), Harry Bunt, Volha Petukhova, and Sander Wubben (Eds.). Association for Computational Linguistics, Stroudsburg, PA, USA, 116-127.

[7] P. Bafna, D. Pramod and A. Vaidya. (2016). Docu- ment clustering: TF-IDF approach. In *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).* Chennai, 2016, pp. 61-66. doi: 10.1109/ICEEOT.2016.7754750

[8] Lau, Jey Han, and Timothy Baldwin. (2016). An Empir- ical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP.* Berlin, Germany, pp. 78–86 arXiv:1607.05368

[9] L.J.P. van der Maaten and G.E. Hinton. (2008). Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov):2579-2605.

[10] L.J.P. van der Maaten. (2009). Learning a Parametric Em- bedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS).* JMLR W&CP 5:384-391.

[11] Kao, A., Poteet, S.R. (2007). *Natural Language Processing and Text Mining.* Springer, London.

[12] Konchady, M. (2006). *Text Mining Application Programming.* Cengage Learning.

[13] Miner, G. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications.* 1st edn. Academic Press.

[14] Aggarwal, C.C., Zhai, C.X. (2012). *Mining Text Data.* pp. 12–14. Springer US.

[15] Beigman Klebanov, B., Knight, K., Marcu, D. (2004). *Text simplification for information seeking applications.* In: *OTM* Meersman, R., Tari, Z. (eds.) LNCS, vol. 3290, Springer, Heidelberg.

[16] Martin, J., Jurafsky, D. (2008). *Speech and language process- ing: An introduction to natural language processing, compu- tational linguistics and speech recognition.* 2nd edn. Prentice Hall.

[17] Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman. (2011). Natural language processing: an intro- duction. *Journal of the American Medical Informatics Associ- ation.* Volume 18, Issue 5, 1 September 2011, Pages 544–551.

[18] D. H. Deshmukh, T. Ghorpade and P. Padiya. *Improving classification using preprocessing and machine learning al- gorithms on NSL-KDD dataset* 2015 International Confer- ence on Communication, Information Computing Technol- ogy (ICCICT), Mumbai, 2015, pp. 1-6. doi: 10.1109/IC- CICT.2015.7045674

[19] Tomas Mikolov, Quoc Le *Distributed Representations of Sentences and Documents Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043*