Open Source Software Development

Project Based Learning Report

Topic Extraction and Customer Opinion Analysis using Deep Learning

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

jiit

विद्या तत्व ज्योतिसमः

Team Members:

1. Sidhant Moza (23303026)
2. Ehtesham Ashraf (23003021)
3. Himani Agrawal (23303025)

Table of Contents

S.No.	Section	Page No.
1	Abstract	2
2	Introduction	2
3	Literature Review	5
3.1	Foundation Theories	8
4	Approach with Development Idea	13
5	Tools, Frameworks & Libraries Required	14
6	Methodology	16
6.1	Preprocessing and Feature Extraction	16
6.2	Clustering of Reviews	19
6.3	Classification using Deep Neural Networks	19
7	Implementation	20
7.1	Data Loading and Preprocessing	22
7.2	Feature Extraction using Word2Vec	24
7.3	Dimensionality Reduction using PCA and t-SNE	26
7.4	Clustering of Reviews	28
7.5	ANN Classifier	32
8	Results and Findings	33
9	Conclusion and Future Scope	38
10	References	39

Abstract

The work presents a novel approach to customer topic modeling, aiming to capture nuanced feedback from customer reviews beyond simple sentiment analysis. By leveraging techniques such as Word2Vec embeddings and clustering algorithms like HDBSCAN, we extract and organize essential information about customer opinions and topics discussed. Our methodology involves preprocessing and feature extraction, followed by clustering of reviews into 15 distinct centroids derived from Word2Vec embeddings. We then employ a deep neural network classifier, enhanced with techniques like SMOTE to address class imbalance, to categorize inputs into these clusters. The classifier architecture, comprising dense layers with Tanh and ReLU activations, effectively distills complex information, demonstrating robust performance in interpreting customer feedback. Through experimentation and optimization of hyperparameters, our approach achieves state-of-the-art accuracy levels, providing actionable insights for product enhancement and performance evaluation based on customer reviews.

Introduction

In today's competitive market landscape, understanding customer opinions and sentiments has become paramount for businesses to thrive. Customer reviews serve as valuable sources of feedback, providing insights into product performance, user satisfaction, and areas for improvement. As customers increasingly rely on online reviews to guide their purchasing decisions, the analysis of these reviews has emerged as a critical component in understanding the customers’ opinion. Among various platforms facilitating customer feedback, Amazon stands out as a prominent hub for software product reviews, hosting a vast repository of unlabelled raw data waiting to be harnessed.

Customer reviews on Amazon and similar platforms carry substantial weight in shaping purchasing decisions. Prospective buyers often rely on these reviews to gauge the quality, usability, and reliability of software products before making informed choices. Moreover, companies leverage customer feedback to enhance their offerings and refine marketing strategies. Amazon hosts an extensive array of software products spanning diverse categories, ranging from productivity tools to entertainment software and beyond. Within this expansive ecosystem, customer reviews guide potential consumers towards or away from specific products based on the collective experiences shared by previous users.

The majority of data available online exists in its raw, unstructured form, without predefined labels or structured formats. This unstructured nature poses a significant challenge for traditional supervised learning approaches, which rely on labeled datasets for training predictive models. Semi-supervised learning emerges as a potent solution to address the inherent limitations posed by raw, unstructured, and unlabelled text data.It facilitates the efficient allocation of labeling resources, minimizing the need for manual annotation of large volumes of data.

Clustering techniques play a pivotal role in semi-supervised learning for text data. The word2vec algorithm, a widely adopted technique in natural language processing, transforms words into high-dimensional numerical vectors that capture their semantic relationships. By applying clustering algorithms to these word2vec features, the review data can be organized into distinct groups based on their thematic similarities, even in the absence of explicit labels.

While traditional clustering algorithms, such as K-Means, struggle with the complex, non-linear structure of textual data, more advanced techniques have proven to be more effective. Density-based clustering algorithms, like DBSCAN and its variant HDBSCAN, can identify clusters of varying sizes, shapes, and densities, better accommodating the unique characteristics of textual data. Additionally, hierarchical clustering methods, such as agglomerative clustering, can capture the inherent hierarchical structure of language, further enhancing the quality of the clustering results.

To address the high dimensionality of the word2vec features, dimensionality reduction techniques like Principal Component Analysis (PCA) and t-SNE (t-Distributed Stochastic Neighbor Embedding) are mostly employed. PCA identifies the most important linear combinations of features which maximize their variance across all data, allowing for a compact representation of the data. In contrast, t-SNE is specifically designed for visualizing high-dimensional data, preserving the local structure of the data while revealing its global structure. By applying these dimensionality reduction techniques, it enables effective clustering and visualization of the customer review data, ultimately facilitating the semi-supervised learning process.

After the successful clustering of customer reviews based on their semantic features, the obtained cluster labels can be used to train machine learning models such as Artificial Neural Networks (ANNs) for the classification of software reviews. By utilizing the inherent structure and patterns revealed through the clustering process, these models will be able to learn and generalize more effectively, even in the absence of extensive labeled data.

This comprehensive approach will empower software companies to better understand their customers, make data-driven decisions, and continuously improve their product offerings, ultimately enhancing customer satisfaction and loyalty.

This report is organized as follows: Section 2 provides a literature review of existing research in customer opinion extraction. Section 3 outlines the project's approach and development ideas. Section 4 goes through the process of developing the project ideas. Section 5 covers the required tools and libraries. Section 6 details the methodology, including text preprocessing, feature engineering, clustering, and semi-supervised learning. Section 7 presents the results and findings. Section 8 concludes the report and discusses future research directions.

Literature Review

Comparative analysis of six relevant research papers :

Reference Paper Name	Paper[1]: Enhancing Depression detection through advance text analytics: integrating BERT, Autoencoder, LSTM models	Paper[2]: Opinion Mining of Consumer Reviews Using Deep Neural Networks with Word-Sentiment Associations	Paper[3]: Toward Identifying Customer Complaints of Cloud Service Providers Using Topic Modelling and Sentiment Analysis	Paper[4]: Understanding Online Customer Touchpoints: A Deep Learning Approach to Enhancing Customer Experience in Digital Retail	Paper[5]: The Use of Topic Modelling in Mining Customers' Reviews	Paper[6]: Analysis of Topic and Sentiment Trends in Customer Reviews Before and After Covid-19 Pandemic
Addressed problem in Literature	Detect depression emotions from text data using BERT & autoencoders over deep LSTM classification model	Extract opinions of customers using text data, DNN model with word sentiment associations over LSTM-CNN layers	Reduced customer satisfaction due to exponentially increasing demand which leads to large text data to be analyzed.	Identification of the main touchpoint's customers value most when shopping online and assess their attitudes towards them	Aims to address the challenges by applying the LDA modeling algorithm to analyze customer reviews such that it improves overall Business Performance.	Aims to understand how customer discussions about headphones have changed, both in terms of topics discussed and sentiments expressed, in context of Pandemic.

Methods or ML/DL models used	Preprocessing(NLP)
	Feature extraction(BERT)
	Feature fusion(AutoEncoders)
	Classification(LSTM)
		Preprocessing
		Autoencoders
		DNN with word sentiment associations(DNN-WSA)
		SoftMax to classify
		measure
			Analyze customer complaints from Microsoft Azure and DigitalOcean cloud platforms.
			BERT Topic for Topic Modelling and VADER (Valence Aware Dictionary and sentiment
			Reasoner) for Sentiment Analysis
				Analysis of Ocado's customer reviews by performing ETM-based topic modeling and sentiment analysis using word2vec & CNN on each topic. Time series analysis is also done afterwards on those topics.	RapidMiner used for Data Preparation, Tokenization to convert text into meaningful words, LDA applied for detecting latent themes in reviews.	Methodology integrates NLP techniques specially LDA for Topic Modelling and SentiWordNet for sentiment analysis.

Performance results/ outcomes	Autoencoder feature fusion is effective in denoising, the model performs reasonably well with relatively lower MAE, RMSE, Rsquared error. Overall structure achieves 98% accuracy.	Word sentiment associations are more effective than word representations based on word embeddings only.	Sentiment analysis on both cloud platforms suggested that Azure had a slightly more proportion of positive feedback than DigitalOcean. Topic Modelling resulted in a total 209 topics out of which 189 were relevant for analysis. Term Score Decline of topics from Azure reviews was less than that of topics from DigitalOcean reviews.	ETM-based topic model resulted in generating 5 topics with superior results than other no. of topics namely "Items", "Online Service", "Quality of food", "Delivery slot" and "Driver". 7 touchpoints were identified.	Provides clear Understanding of the prevalent themes in customer reviews to enhance specific aspects of their meal delivery services and overall customer satisfaction.	By utilizing LDA, successfully identifies Six important customer reviews on Headphones: Durability Issues, Usage Contexts, Noise Cancellation, Features, Quality, and Customer Service.

The literature review highlights Hajek, P. et. al.[2] work on customer review analysis, which involves processing raw customer feedback into machine-understandable forms and extracting opinions while maintaining contextual relations. The author employs a deep neural network model with word sentiment associations for this task. The review underscores the significance of analyzing customer opinions to gather essential feedback. It critiques existing sentiment analysis methods, noting their tendency to oversimplify reviews into basic categories like good/bad/neutral, thus potentially missing crucial points for product improvement. An example is provided to illustrate how traditional sentiment analysis might overlook valuable suggestions for product enhancement embedded within positive reviews.

In the realm of sentiment analysis and customer feedback interpretation, recent studies have showcased innovative approaches leveraging advanced text analytics techniques and deep learning architectures [1][2]. For instance, Firoz et. al.[1] presents an approach that integrates BERT, autoencoders, and LSTM models to enhance depression detection through text analysis. Similarly, Hajek et. al.[2] employs deep neural networks with word-sentiment associations to extract opinions from consumer reviews, demonstrating the effectiveness of such methodologies in understanding customer sentiment.

As the volume of text data generated by customers continues to grow exponentially, there arises a need to efficiently analyze and extract insights from this data [3]. This challenge is addressed in paper by Alghamdi et. al.[3], which focuses on identifying customer complaints of cloud service providers using topic modeling and sentiment analysis, aiming to improve overall business performance.

Furthermore, research efforts have been directed towards understanding customer touchpoints and their impact on the overall customer experience in digital retail [4]. Yilin et. al.[4] employs a deep learning approach to identify the main touchpoints valued by customers when shopping online, providing valuable insights for enhancing customer satisfaction.

Additionally, the use of topic modeling techniques, such as Latent Dirichlet Allocation (LDA), has been instrumental in uncovering prevalent themes in customer reviews and sentiments expressed [5][6]. Letter et. al.[5] utilizes LDA for topic modeling and sentiment analysis to understand customer discussions about headphones, particularly in the context of the COVID-19 pandemic. While Suryadi et. al.[6] in their work apply LDA for topic modeling and sentiment analysis on Ocado's customer reviews, identifying key themes and conducting time-series analysis to track changes over time.

These studies highlight the importance of advanced text analytics methodologies, including natural language processing (NLP) techniques and deep learning architectures, in gaining insights from customer feedback. By leveraging these approaches, businesses can enhance their understanding of customer sentiment, identify areas for improvement, and ultimately, enhance customer satisfaction and loyalty.

Foundation Theories

Word2Vec

Word2Vec is a popular technique in natural language processing that learns low-dimensional, continuous vector representations of words, also known as word embeddings. The word2vec model is based on the distributional hypothesis, which states that words with similar meanings tend to appear in similar contexts.
Mathematically, the word2vec model aims to learn a function f: V → ℝ^d that maps each word w in the vocabulary V to a d-dimensional vector representation f(w). The model is trained to maximize the conditional probability of observing a target word given its surrounding context words, or vice versa.
There are two main architectures for the word2vec model:
Continuous Bag-of-Words (CBOW): The CBOW model predicts the current word based on the context (surrounding words). Mathematically, the objective function is to maximize the log-likelihood:

$$\sum log\, P(w_t \,|\, w_{t-c}, \dots , w_{t+c})$$

where w_t is the target word and $w_{t-c}, \dots , w_{t+c}$ are the context words within a window of size c.

Skip-Gram: The Skip-Gram model predicts the surrounding context words given the current word. Mathematically, the objective function is to maximize the log-likelihood:

$$\sum\sum_{j} log\, P(w_{t+j} \mid w_t)$$

where w_t is the target word and w_{t+j} are the context words.

These conditional probabilities are typically modeled using the softmax function:

$$P(w_o|w_i) = \frac{e^{v_o^T \cdot v_i}}{\sum_w e^{v_w^T \cdot v_i}}$$

where v_o and v_i are the output and input vector representations of the words, respectively, and the sum in the denominator is taken over all words in the vocabulary.

The trained word2vec model provides a rich, semantic representation of words, where words with similar meanings are positioned close to each other in the vector space. These word embeddings can then be used as input features for various natural language processing tasks, such as text classification, sentiment analysis, and named entity recognition.

Dimensionality Reduction Techniques

1. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique which aims to transform a high-dimensional dataset into a lower-dimensional representation while preserving the maximum amount of variance present in the original data. This process is particularly valuable when dealing with text data, as it can help overcome the challenges posed by the inherent high dimensionality of textual features, such as those derived from the Bag-of-Words (BoW) or word embedding models. PCA seeks to find a set of orthogonal vectors, known as principal components, that capture the directions of maximum variance in the data.

Let X be a dataset consisting of n samples and p features, represented as an n×p matrix. Let Z be the matrix of principal components, where each column represents a principal component. The transformation from the original data space X to the principal component space Z is given by:

$$Z=XV$$

where V is the matrix of eigenvectors of the covariance matrix of X.
The eigenvectors represent the directions of maximum variance, and the corresponding eigenvalues represent the amount of variance explained by each principal component.
The covariance matrix C of X is computed as:

$$C = \frac{1}{n-1}(X - \overline{X})^T (X - \overline{X})$$

where $\overline{X}$ is the mean vector of X computed column-wise.

These principal components form a new coordinate system, where the first principal component accounts for the largest possible variance, the second principal component accounts for the second-largest variance, and so on. By applying PCA to the embeddings of text data, researchers can effectively reduce the dimensionality of the feature space, mitigate the impact of sparsity, and extract the most informative latent features for subsequent analysis and modeling.

2. t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a powerful technique for nonlinear dimensionality reduction and visualization of high-dimensional data in low-dimensional space. Unlike linear methods such as PCA, t-SNE constructs a probability distribution over pairwise similarities in the high-dimensional space and a similar distribution in the low-dimensional space. It minimizes the Kullback-Leibler (KL) divergence between these two distributions to find an optimal low-dimensional representation of the data.

Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of high-dimensional samples. The probability $p_{j|i}$ that a data point x_i would pick x_j as its neighbor in the high-dimensional space is defined as a conditional probability that is proportional to the similarity between those data points:

$$p_{j|i} = \frac{exp\left(-\frac{||x_i - x_j||^2}{2\sigma_i^2}\right)}{\sum_{k\neq i} exp\left(-\frac{||x_i - x_k||^2}{2\sigma_i^2}\right)}$$

where σ_i^2 is the variance of the Gaussian kernel for x_i, which is determined using a perplexity parameter. The probability p_{ij} is then calculated as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

Similarly, the probability p_{ij} that a data point y_i would pick y_j as its neighbor in the low dimensional space is defined using a t-distribution with one degree of freedom:

$$q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_k \sum_{l\neq k} (1+||y_k - y_l||^2)^{-1}}$$

t-SNE minimizes the KL divergence between the distributions $P = \{p_{ij}\}$ and $Q = \{q_{ij}\}$ using gradient descent optimization. The cost function to be minimized is given by:

$$KL(P||Q) = \sum_{i\neq j} p_{ij} log\left(\frac{p_{ij}}{q_{ij}}\right)$$

t-SNE iteratively adjusts the low-dimensional embeddings $Y = \{y_1, y_2, \ldots, y_N\}$ to minimize this cost function. t-SNE has become popular for visualizing high-dimensional datasets due to its ability to reveal complex structures and clusters present in the data. However, it is computationally intensive and sensitive to its hyperparameters, requiring careful tuning for optimal performance.

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is an extension of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm that automatically determines the number of clusters and identifies noise points in a dataset. HDBSCAN constructs a hierarchical representation of clusters by considering different levels of cluster granularity, providing insights into the data's inherent structure.

The HDBSCAN algorithm is an extension of both DBSCAN and OPTICS. While DBSCAN operates under the assumption that the criteria for clustering is global homogeneity, meaning it may face challenges in effectively identifying clusters with varying densities, HDBSCAN addresses this limitation by constructing an alternative representation of the clustering problem, which enables exploration of all potential density scales, ensuring a more comprehensive capture of clusters with diverse densities.

HDBSCAN first defines the core distance $d_c(x_i)$ of a sample x_i as follow:

$$d_c(x_i) = distance(x_i, x_m)$$

where x_m is the m^{th}-nearest neighbor to x_i, counting itself. The parameter ‘m’ is the minimum no. of samples allowed within its core distance and can be tuned according to the problem. According to this, it also defines the minimum core distance ε.

It then defines $d_r(x_i, x_j)$ as the mutual reachability distance between 2 points x_i, x_j as follows:

$$d_r(x_i, x_j) = max\{ d_c(x_i), d_c(x_j), d(x_i, x_j)\}$$

A mutual reachability graph G is defined by associating each sample as a vertex and the mutual reachability distances between the samples as the edges. Out of these, the vertices corresponding to points whose core distances are less ε, are declared as outliers or noise. The edges whose values are greater than ε are also removed. The remaining points are then clustered by constructing a Minimum Spanning Tree (MST) using a hierarchical approach considering the mutual reachability distances between the points.

Text Clustering

Text clustering is a fundamental unsupervised learning technique employed to organize unstructured textual data into coherent groups or clusters based on the inherent similarities within the corpus. This process aims to uncover hidden patterns, themes, and relationships that exist in large collections of text documents, such as customer reviews, news articles, or research papers.

At the core of text clustering lies the fundamental assumption that documents discussing similar topics or expressing related sentiments will exhibit a higher degree of textual similarity compared to those addressing disparate subject matters. The primary objective of text clustering algorithms is to minimize the intra-cluster distance while maximizing the inter-cluster distance

To achieve this objective, text clustering algorithms typically involve the following key steps: (1) text preprocessing, (2) feature extraction, (3) dimensionality reduction, and (4) cluster formation. During the preprocessing stage, the raw textual data is cleaned and normalized, often involving tokenization, stopword removal, and stemming or lemmatization. Feature extraction techniques, such as the Bag-of-Words (BoW) model, Term Frequency-Inverse Document Frequency (TF-IDF), or word embedding methods (e.g., Word2Vec, GloVe), are then employed to transform the text data into a numerical representation suitable for clustering.

Approach with Development Idea

Topic identification through a literature review of research papers is a critical process in academic research that involves systematically analyzing existing scholarly works to identify prevalent themes, gaps in knowledge, and areas requiring further investigation. The literature review serves as the foundation for refining research objectives, guiding the formulation of research questions, and providing a comprehensive understanding of the current state of knowledge in a particular field. By reviewing relevant research papers, scholars can describe the key concepts, methodologies, and findings that have shaped the discourse on a specific topic. This process not only aids in avoiding redundancy but also facilitates the identification of emerging trends and unresolved issues.

Customer review analysis using deep learning is the main objective function, while after going through multiple research papers a comparative analysis of each research paper was formulated. From these 6 research papers we could identify the below candidate specific research objectives as mentioned below :

- Topic analysis
- Opinion review
- Customer Emotion detection
- Customer sentiment analysis post COVID-19

On reviewing further about the candidate objectives mentioned, we identified that the topic extraction and opinion analysis of customer reviews is the most relevant and novice field to work on. On further reviewing the existing literature, topic analysis sees a wide variety of applications in real world analytics.

Few of major present applications of finalized topic (topic extraction and analysis of customer reviews):

Classifying the age group of the customer from text response

Detecting the overall satisfaction of the customers for each product

Summarizing the customer reviews for each product

Extracting the topic of customer reviews (frequent words analysis)

Customer response sentiment analysis pre & post COVID-19 and many more such..

The topic extraction and analysis build a foundation for the analysis which can further have more detailed applications which can help scientists to build multiple decision model frameworks.

Tools, Frameworks and Libraries Used

In the pursuit of extracting meaningful insights from customer reviews, our project uses a dataset obtained from UC San Diego, consisting of 12,805 customer reviews (ratings, reviews, votes, summaries) for software products, sourced from Amazon. The initial phase involves meticulous preprocessing using Natural Language Processing techniques to refine and clean the extracted data. Python is chosen as the primary programming language for its versatility and extensive libraries in the realm of data science.

For the critical task of opinion extraction, we plan to employ cutting-edge techniques, namely Word2Vec text embedding, clustering techniques such as KMeans, Agglomerative, BIRCH, HDBSCAN, and Artificial Neural Networks for classification. These methods ensure a comprehensive understanding of customer sentiments and preferences. The integration of advanced deep learning models and Python's rich ecosystem positions our project to yield insightful analyses and contribute to the field of sentiment analysis and topic extraction from customer reviews.

Identified tools for the project :

- Google colab
- Dataset
- Cloud Drive
- Python
- Jupyter Notebook
- Git and Github
- Numpy and Pandas
- Seaborn
- SciKit and PyTorch
- Keras
- NLTK and Spacy
- Word2Vec
- Gensim
- SkLearn
- Draw.io
- YAKE
- PCA and t-SNE
- KMeans
- SMOTE
- HDBSCAN
- Keras

Methodology

In our work, we introduce customer topic modeling, which preserves and depicts customer points of improvement and subjects discussed through relational clusters. Inspired by the need to capture specific feedback related to product enhancements and performance issues, our approach extracts vital information about customer opinions and topics discussed. The cluster centroids represent distinct topics within the reviews, condensing relevant information and enabling the machine to link closely related topics to the query. This advancement moves beyond traditional sentiment analysis towards a more nuanced topic modeling approach for customer reviews.

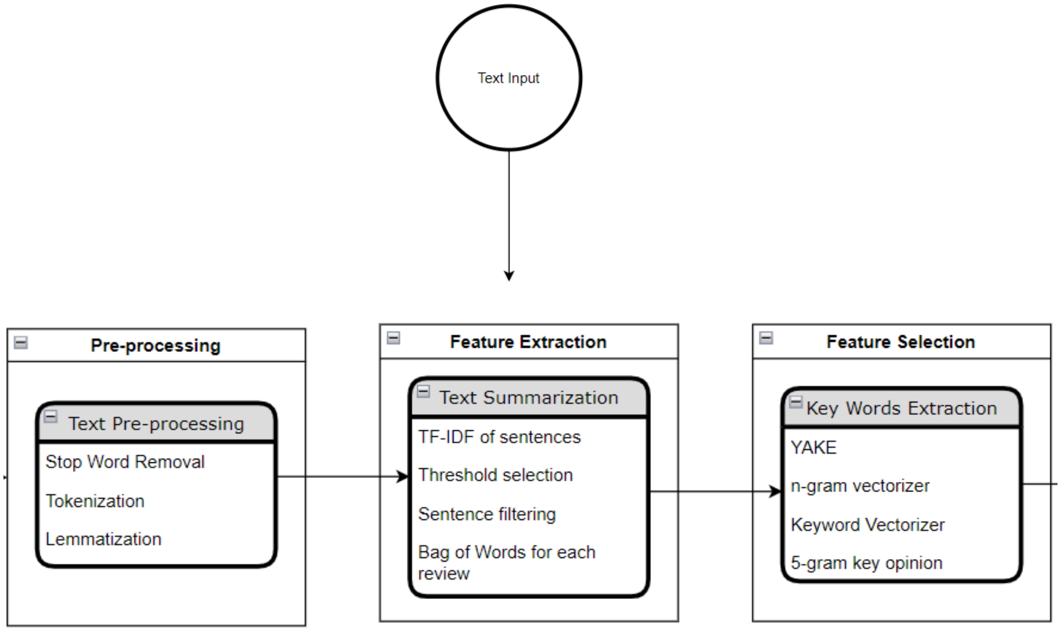
Working on raw data about customer reviews needs a bit of a different approach, the customer reviews contains a lot of information regarding specific and general characteristics about the product in discussion. A naive approach would be to build around a classifier which simply suggests if a review is positive or negative or neutral. But customer reviews are much more than that, many times they consist of points of product improvement, embedded within a negative review, while a positively classified review would also give a short message about product nitti-gitties.

Here in this approach we are clustering all reviews broadly in 15 clusters to obtain their centroids as review labels. On obtaining the review labels, a deep classifier is run over 15 classes to train over the class labels. Clusters are not obtained as a simple algorithm, but derived from Word2Vec encodings preserving their contextual relation between words and sentences.

The research methodology is divided into multiple stages as follows:

I. Preprocessing and Feature Extraction

The steps are explained in detail along with the help of a flow diagram:

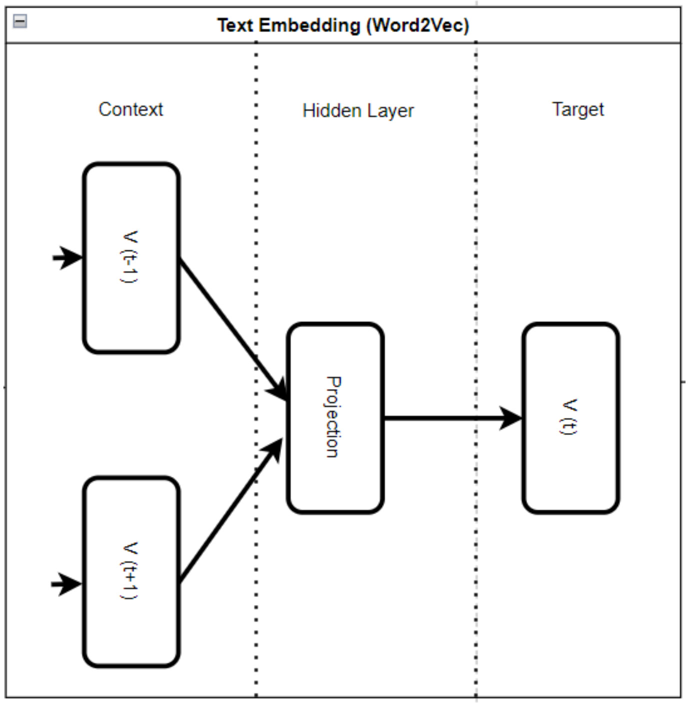


Here we provide the text input to the machine just as raw text input, this raw text goes through detailed steps such as:

1. Preprocessing
2. Feature Extraction
3. Feature Selection

The data pre-processing involves cleansing, transforming, and organizing data to enhance its quality and usability. One crucial aspect of data preprocessing is handling missing values. Techniques such as imputation or deletion of incomplete records are employed to mitigate the impact of missing data on subsequent analyses. Additionally, outlier detection and removal help in ensuring the robustness of the dataset by eliminating anomalous data points that may skew the results. Moreover, data normalization or scaling is performed to standardize the range of features, thereby facilitating fair comparison between different variables. Through meticulous data preprocessing, noise is reduced, and the underlying patterns within the data become more discernible.

This is followed by the feature extraction from the data. Feature extraction involves identifying and selecting relevant attributes or characteristics from the raw data that encapsulate the essential information for analysis. This process is crucial for dimensionality reduction and enhancing the efficiency of subsequent modeling tasks. Feature selection methods, such as forward selection or backward elimination, help in identifying the most informative subset of features, thereby simplifying the analysis without compromising its effectiveness. Additionally, domain-specific knowledge enables the creation of tailored features that encapsulate domain intricacies, thereby enriching the dataset with meaningful insights.

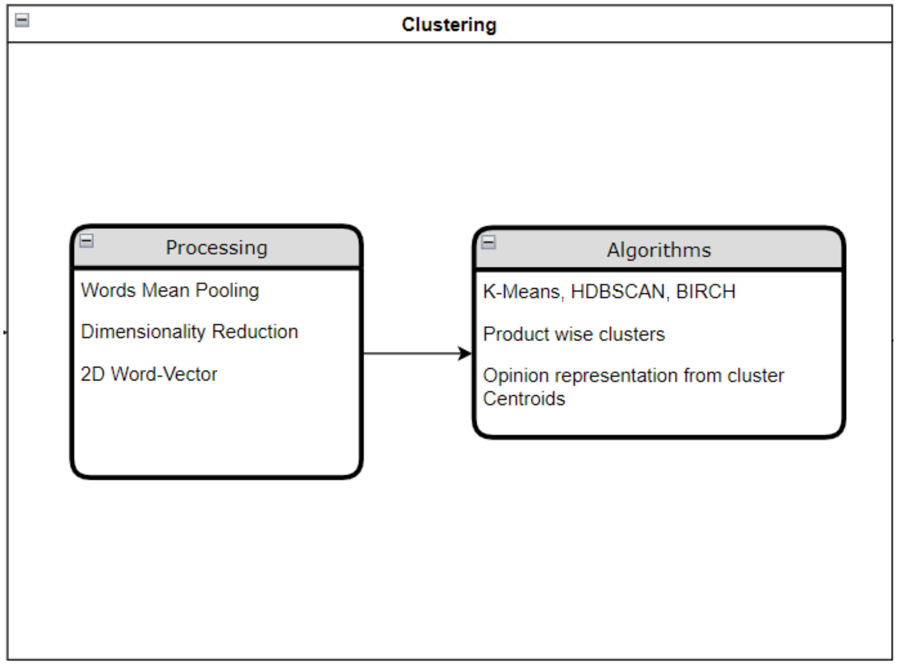


These selected features are extracted and then provided to a text vectorizer known as Word2Vec. Word2Vec captures semantic similarities between words, enabling nuanced understanding of language semantics. Words with similar meanings are mapped to nearby regions in the vector space, facilitating tasks like word similarity and analogy detection. Word2Vec embeddings exhibit compositionality, allowing the combination of word vectors to represent phrases or sentences.

Implementing Word2Vec for vectorizing raw text involves several steps. First, a corpus of text data is preprocessed to remove noise, tokenize sentences, and construct a vocabulary. Next, the Word2Vec model is instantiated with hyperparameters such as vector dimensionality, window size, and training algorithm. The model is then trained on the preprocessed corpus using stochastic gradient descent or other optimization techniques. During training, the model updates the word vectors iteratively to minimize the loss function, thereby improving the embeddings' quality. Once trained, the word vectors can be used to represent words in downstream natural language processing tasks such as sentiment analysis, document classification, or machine translation.

II. Clustering of Reviews

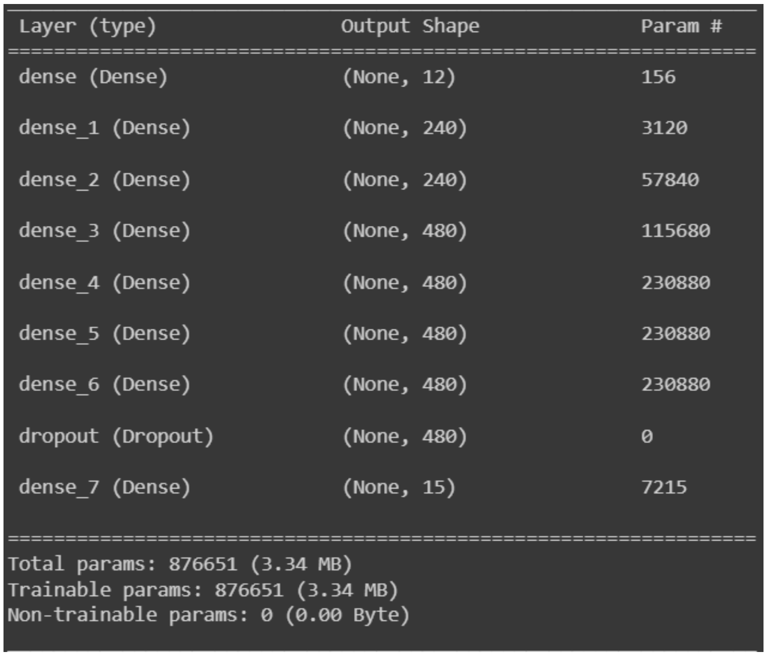
To obtain the data labels and to better understand the text representations, all text is clustered in multiple clusters. To obtain relational clusters, various algorithms such as KMeans, HDBSCAN, BIRCH are implemented. K-means is a classic clustering algorithm that partitions data points into K clusters based on their Euclidean distances to cluster centroids. In the context of textual data, each document or word is represented as a vector using techniques like Word2Vec or TF-IDF. K-means then iteratively assigns data points to the nearest centroid and updates the centroids based on the mean of the assigned points. On the other hand, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that identifies clusters of varying shapes and densities. It constructs a hierarchy of clusters and determines the optimal clustering solution based on the stability of the clusters' densities.



III. Classification using Deep Neural Network

The data and its content is now in machine readable state along with all the relevant information available, this stream is passed to a deep neural network to train it as a classifier. The deep architecture consists of neurons arranged in successive layers, with each neuron connected to every neuron in the adjacent layers. Implementing dense layers with Tanh and Rectified Linear Unit (ReLU) activation functions introduces non-linearity into the network, enabling it to learn and model intricate relationships within the data. ReLU activation function sets the output of a neuron to zero for negative inputs and preserves positive inputs, effectively introducing a thresholding mechanism. This non-linear activation fosters better representation learning and enables ANNs to approximate complex functions. While on the other side Tanh activation function pushes the output towards -1 or +1 giving it a wider perspective mapping. The Tanh activation function is powerful with its property of zero centered range of output nonlinearity mapping. Consequently, dense layers with Tanh activation functions serve as powerful building blocks in the construction of deep neural networks capable of tackling diverse machine learning tasks, from image classification to natural language processing. Towards each hidden layer a dropout layer is added. Dropout is a regularization technique that operates by randomly deactivating a fraction of neurons during training, effectively preventing co-adaptation of neurons and promoting model robustness. Complete layer wise details can be seen from the below architecture. It can be seen that the trainable parameters are almost 1 million.

During the cluster dataset analysis, it was noticed that the clusters exhibited significant variance in support count, with some containing a large number of points and others representing minority clusters. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. SMOTE effectively increases and balances the number of points within minority clusters, thereby reducing bias and ensuring that all clusters contribute meaningfully to the model's training process.



```
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 12)                156

 dense_1 (Dense)             (None, 240)               3120

 dense_2 (Dense)             (None, 240)               57840

 dense_3 (Dense)             (None, 480)               115680

 dense_4 (Dense)             (None, 480)               230880

 dense_5 (Dense)             (None, 480)               230880

 dense_6 (Dense)             (None, 480)               230880

 dropout (Dropout)           (None, 480)               0

 dense_7 (Dense)             (None, 15)                7215

=================================================================
Total params: 876651 (3.34 MB)
Trainable params: 876651 (3.34 MB)
Non-trainable params: 0 (0.00 Byte)
```

At its core, the classifier architecture consists of an input layer followed by several hidden layers, a dropout layer, and an output layer. The input layer, with 12 units, serves as the entry point for the data, receiving input and passing it forward for processing. It's the starting point for the network's journey through the data.

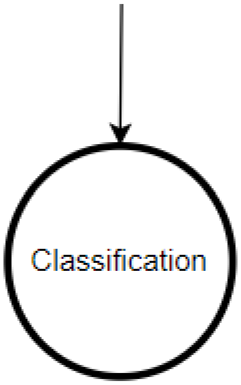
The hidden layers, which perform the bulk of the computational work, come next. Ranging from layers with 240 units to those with 480 units, these dense layers are responsible for transforming the input data through nonlinear operations. Each layer learns increasingly complex features and patterns, extracting valuable information to aid in the classification task.

A dropout layer follows the hidden layers, acting as a regularization mechanism to prevent overfitting. During training, it randomly drops a fraction of input units, forcing the network to learn more robust and generalizable features. This helps improve the model's performance on unseen data and enhances its ability to generalize to new examples.

Finally, the output layer, comprising 15 units, serves as the endpoint of the network. It produces the final predictions, assigning probabilities to each class in the classification task. The softmax activation function typically accompanies this layer, converting the raw output scores into probability distributions, making it easier to interpret the model's predictions.

The complete deep neural network structure architecture boasts a formidable 876,651 parameters while working as an ANN, highlighting the complexity and capacity of the model to learn from data. While all parameters contribute to the network's functionality, it's essential to note that not all are trainable during the training process. The trainable parameters, however, undergo adjustments to optimize the model's performance and enhance its ability to classify inputs accurately.

This model is trained and performs decently well on all performance metrics. Hence a strong classifier is developed along with the strength of providing the customer opinions specifically about the relevant products. This not only empowers this deep model to make classifications but also gives the overall flow strength to present product specific customer opinions with relevant information.

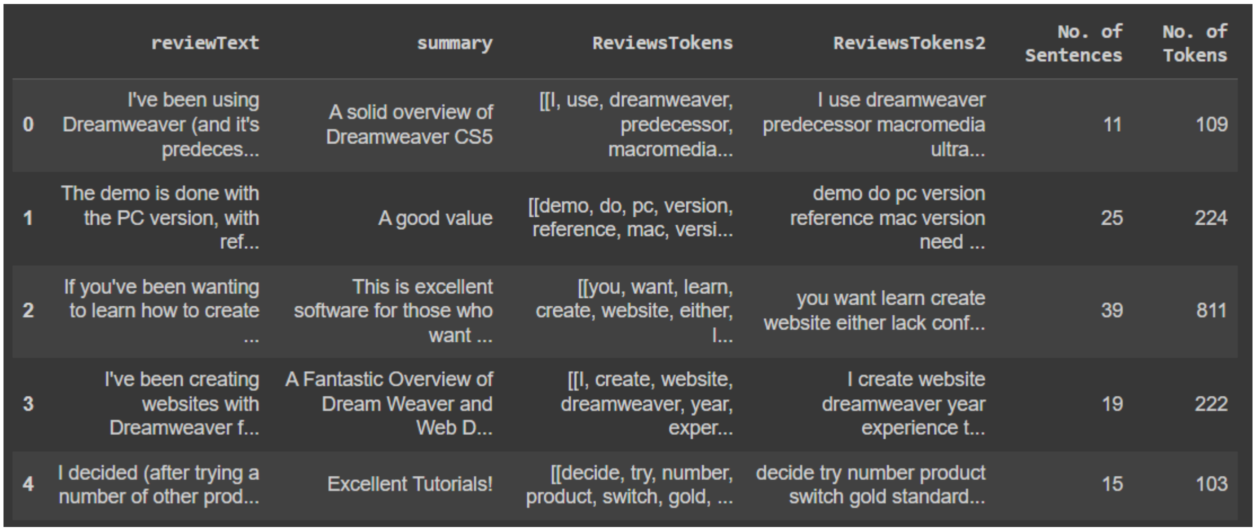


Implementation

a) Data Loading and Preprocessing

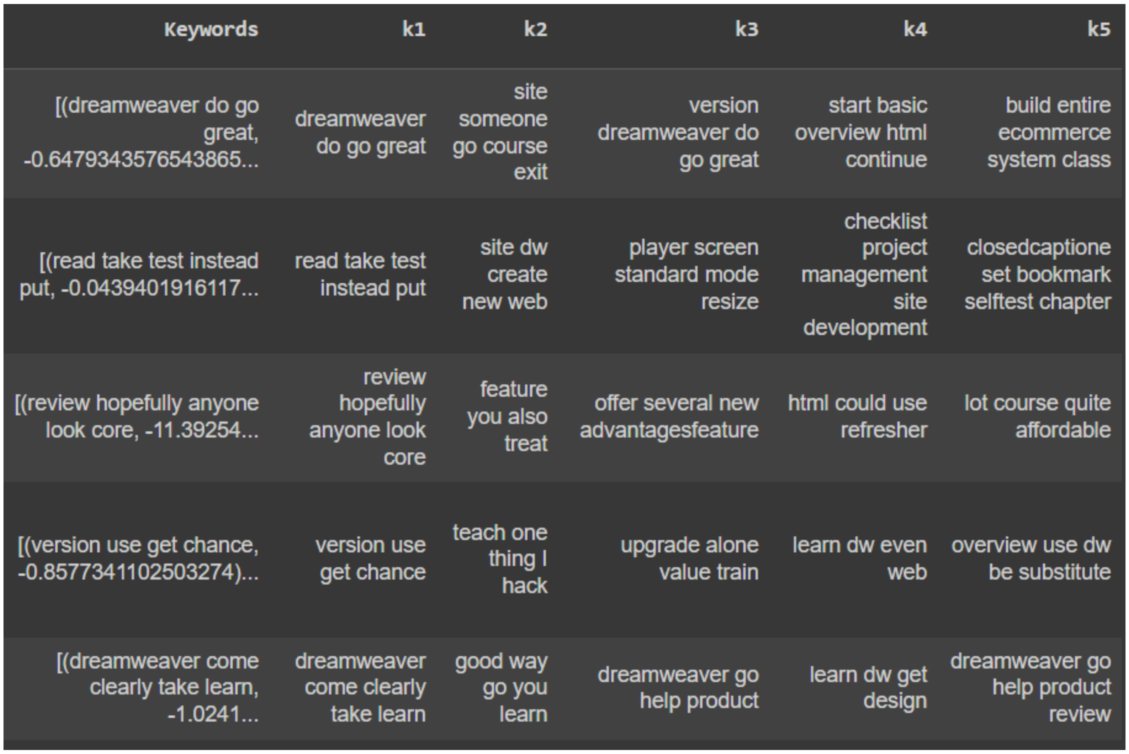
The process of implementation began with data loading in raw ".gz" compressed format. The data was extracted from it and formatted in terms of a dataframe. The columns which were relevant for our project such as 'reviewText' and 'summary' were kept and the rest were discarded.

Then, a pipeline of preprocessing techniques which consisted of stopword removal, punctuation marks removal, elimination of short reviews, tokenization, lemmatization was applied on the 'reviewText' column to get 2 new preprocessed reviews columns. The reviews in the 'ReviewTokens' column were stored in form of tokenized sentences and tokenized word as list of list. This was done for it to be later used for Word2Vec embedding tasks which require input as sentence and word tokenized. The 'ReviewTokens2' column stored the same preprocessed reviews but without any tokenization for it to be later used for Keyword Extraction using YAKE. Also, other features such as no. of sentences and no. of tokens were extracted from the reviews. The preprocessed reviews in form of a dataframe table is shown below:



	reviewText	summary	ReviewsTokens	ReviewsTokens2	No. of Sentences	No. of Tokens
0	I've been using Dreamweaver (and it's predeces...	A solid overview of Dreamweaver CS5	[[I, use, dreamweaver, predecessor, macromedia...	I use dreamweaver predecessor macromedia ultra...	11	109
1	The demo is done with the PC version, with ref...	A good value	[[demo, do, pc, version, reference, mac, versi...	demo do pc version reference mac version need ...	25	224
2	If you've been wanting to learn how to create ...	This is excellent software for those who want ...	[[you, want, learn, create, website, either, l...	you want learn create website either lack conf...	39	811
3	I've been creating websites with Dreamweaver f...	A Fantastic Overview of Dream Weaver and Web D...	[[I, create, website, dreamweaver, year, exper...	I create website dreamweaver year experience t...	19	222
4	I decided (after trying a number of other prod...	Excellent Tutorials!	[[decide, try, number, product, switch, gold, ...	decide try number product switch gold standard...	15	103

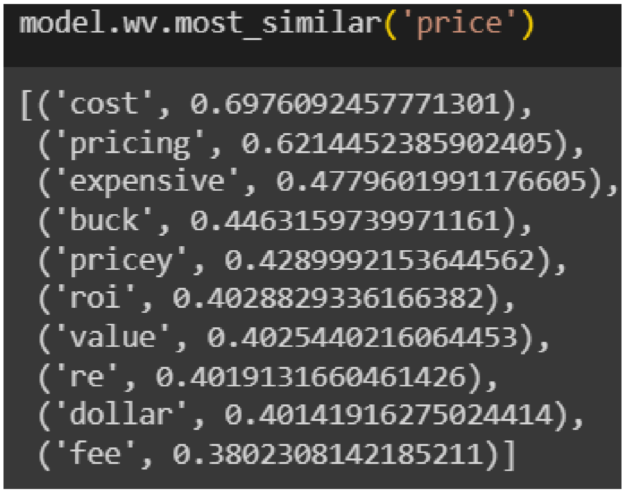
Important keywords from the reviews were extracted using the YAKE library. The maximum n-gram extracted was 5, deduplication threshold set to 0.9, number of sets of keywords to be extracted were 20. Hence, for each review, 20 sets of 5 keywords each were extracted and the best among them were chosen on the basis of a score attached with them. Top 5 sets out of 20 were chosen for each review and stored in 5 different columns. The dataframe storing all the keywords is shown below:



Keywords	k1	k2	k3	k4	k5
[(dreamweaver do go great, -0.6479343576543865...	dreamweaver do go great	site someone go course exit	version dreamweaver do go great	start basic overview html continue	build entire ecommerce system class
[(read take test instead put, -0.0439401916117...	read take test instead put	site dw create new web	player screen standard mode resize	checklist project management site development	closedcaptione set bookmark selftest chapter
[(review hopefully anyone look core, -11.39254...	review hopefully anyone look core	feature you also treat	offer several new advantagesfeature	html could use refresher	lot course quite affordable
[(version use get chance, -0.8577341102503274)...	version use get chance	teach one thing I hack	upgrade alone value train	learn dw even web	overview use dw be substitute
[(dreamweaver come clearly take learn, -1.0241...	dreamweaver come clearly take learn	good way go you learn	dreamweaver go help product	learn dw get design	dreamweaver go help product review

b) Feature Extraction using Word2Vec

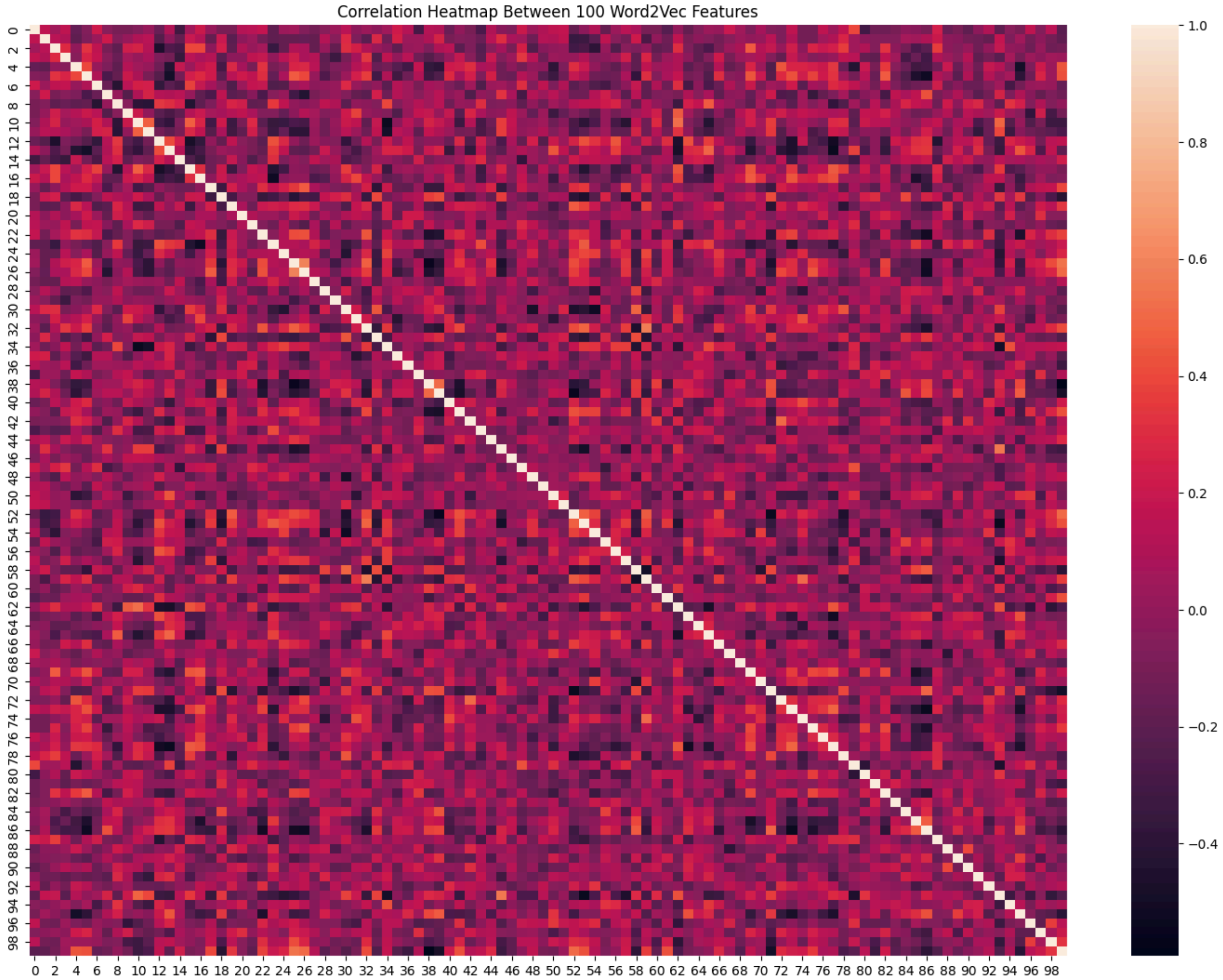
A Word2Vec model was initialized with context window size as 10 and vector size as 100. The preprocessed reviews in the column ‘ReviewTokens2’ were given as input to the Word2Vec model. The model was trained for 100 epochs and then its performance was checked by finding how closely it stores words with similar meanings. The results as shown below depict that it was trained well on the reviews.



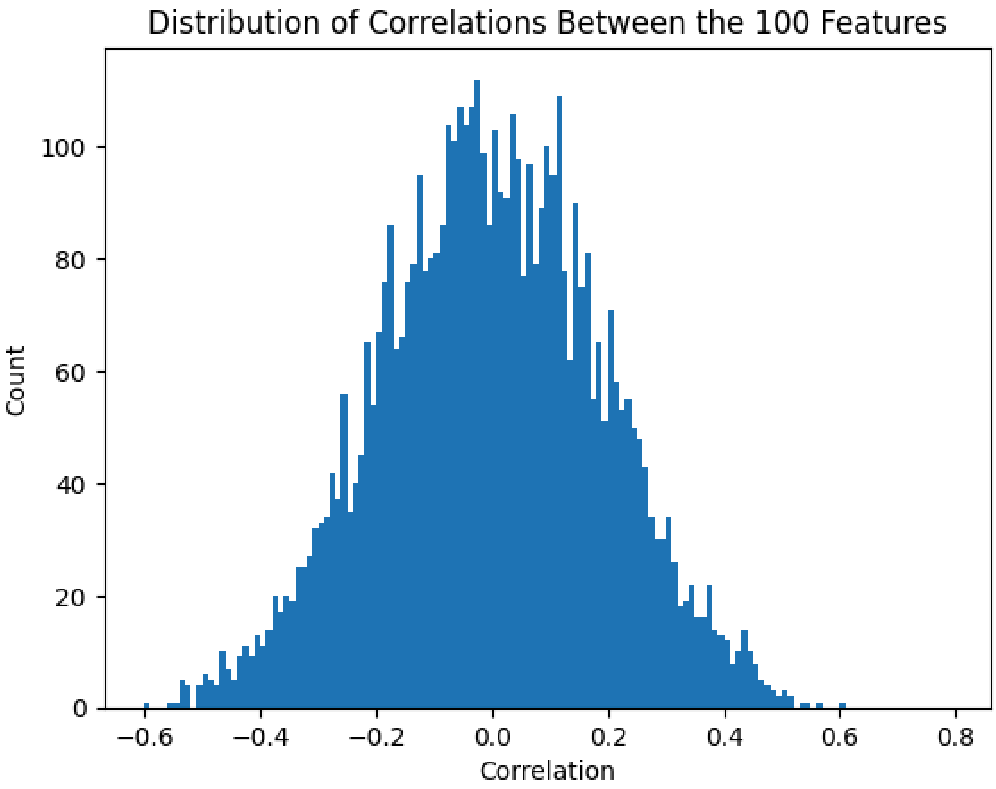
```
model.wv.most_similar('price')

[('cost', 0.6976092457771301),
 ('pricing', 0.6214452385902405),
 ('expensive', 0.4779601991176605),
 ('buck', 0.4463159739971161),
 ('pricey', 0.4289992153644562),
 ('roi', 0.4028829336166382),
 ('value', 0.4025440216064453),
 ('re', 0.4019131660461426),
 ('dollar', 0.40141916275024414),
 ('fee', 0.3802308142185211)]
```

The 100 features extracted using Word2Vec were analyzed using correlation between them. A heatmap was plotted as shown below.

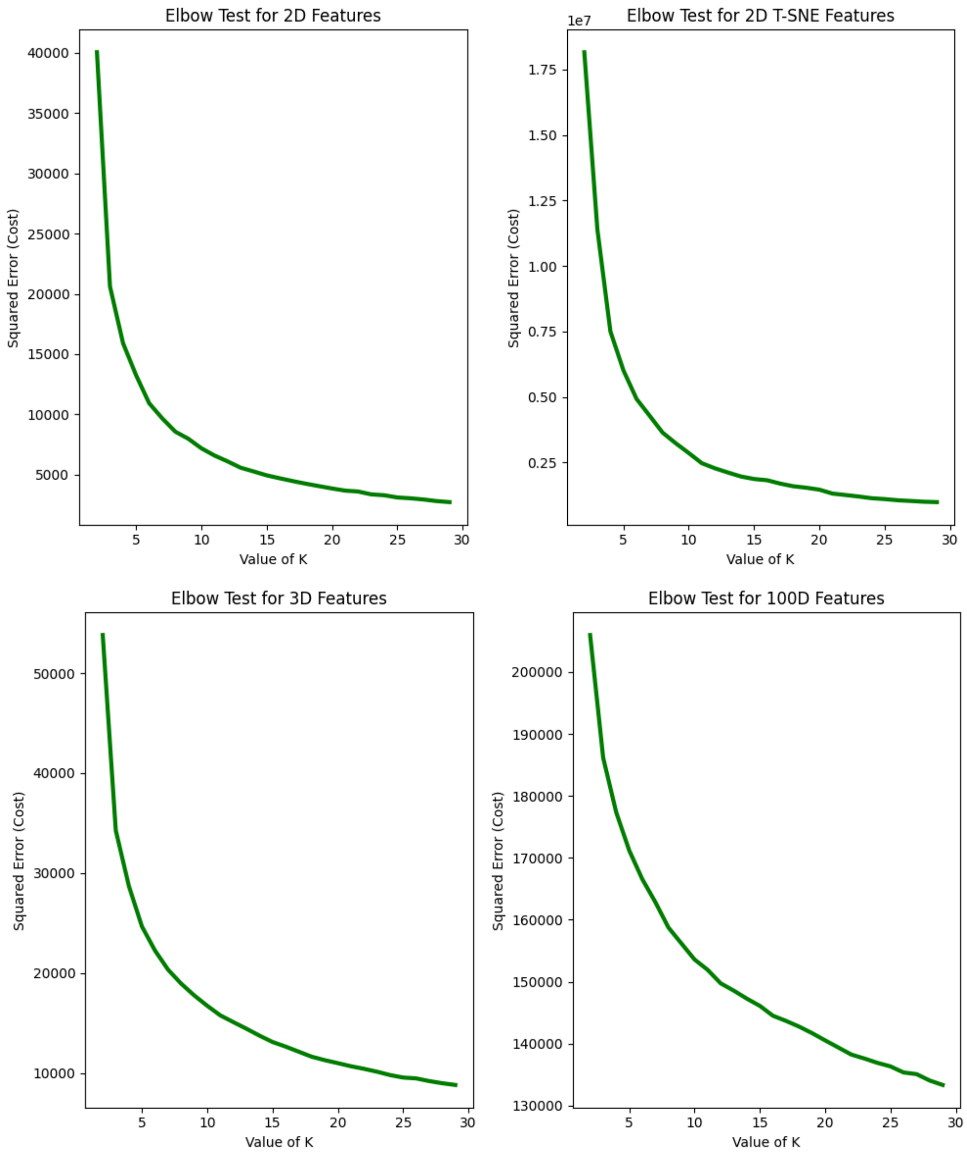


To understand better about the distribution of the correlations between the features, a histogram was plotted as shown below. We observed that most of the features had near 0 correlation between them and no pair of features out of 100 features had correlation values higher than 0.6, hence the Word2Vec model was successful in extracting very diverse features out of the reviews.

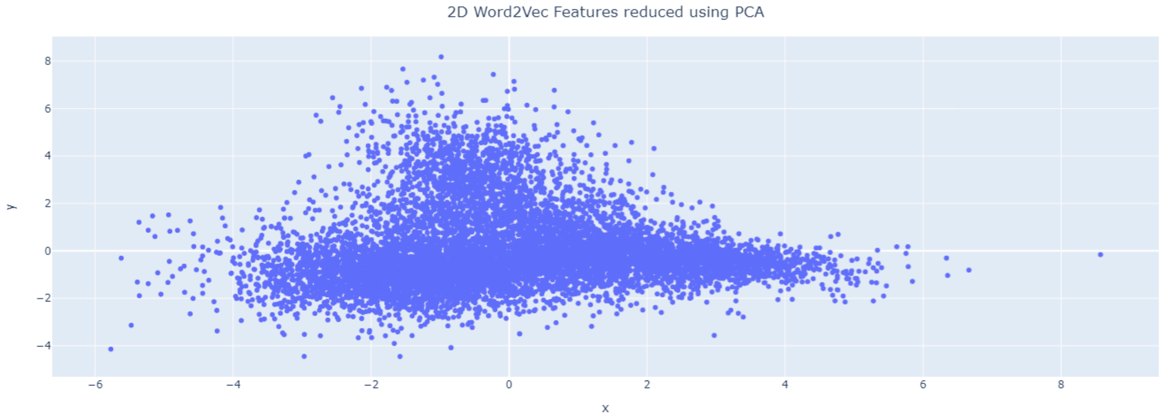


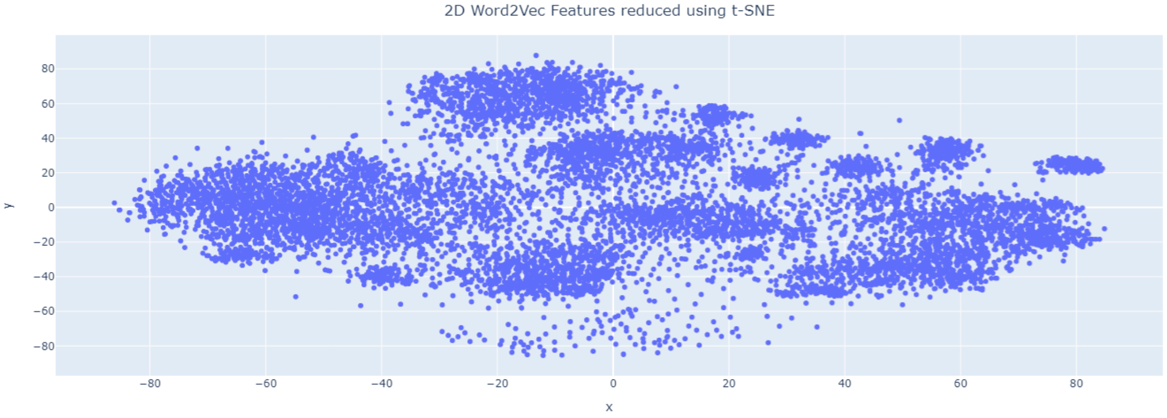
c) Dimensionality Reduction using PCA and t-SNE

The dimensionality of these 100-dimensional features was reduced down at different dimensions using PCA and t-SNE. The reduced features were tested for the number of clusters by performing the Elbow Test using K-Means clustering algorithm. The results, shown below, depict that the curve in elbow test in higher dimensions such as 100D do not show any elbow point due to the curse of dimensionality. Hence, we chose to reduce the features down to 2D. It was also observed that the best values of the number of clusters lied between 10-15.



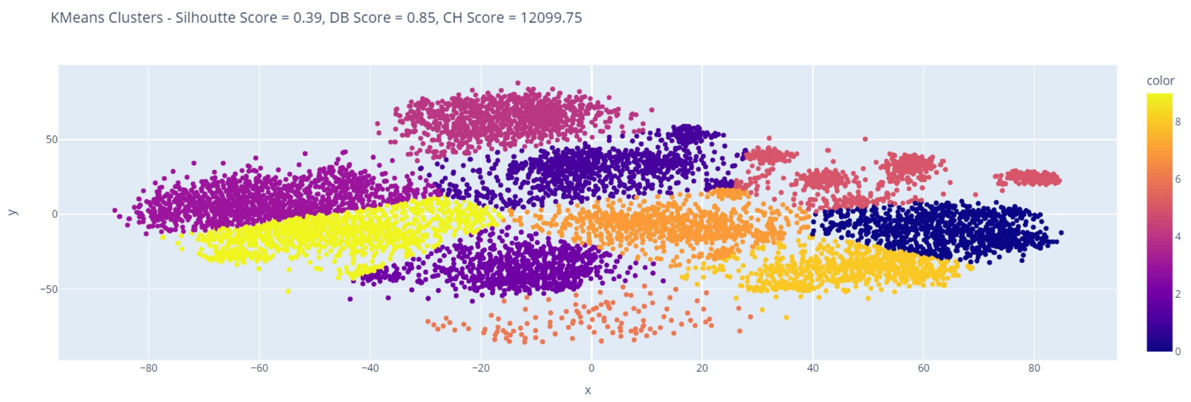
The choice of technique to be used for dimensionality reduction among PCA and t-SNE was taken by visualizing the 2D features obtained from both. The plot of PCA features showed one big cluster with outliers, while the plot of t-SNE features showed many clusters varying in size and density. Hence, the features from t-SNE were chosen for further analysis.



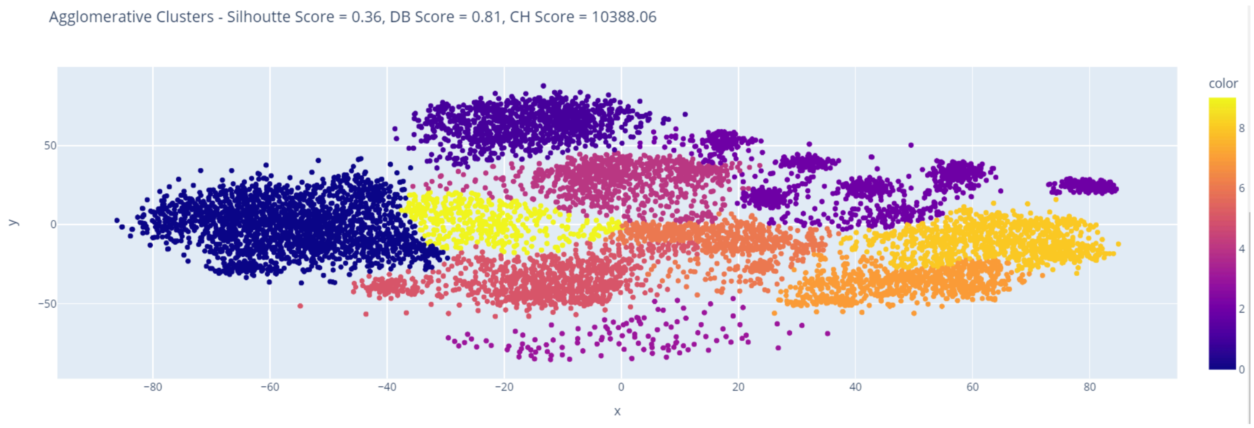


d) Clustering of Reviews

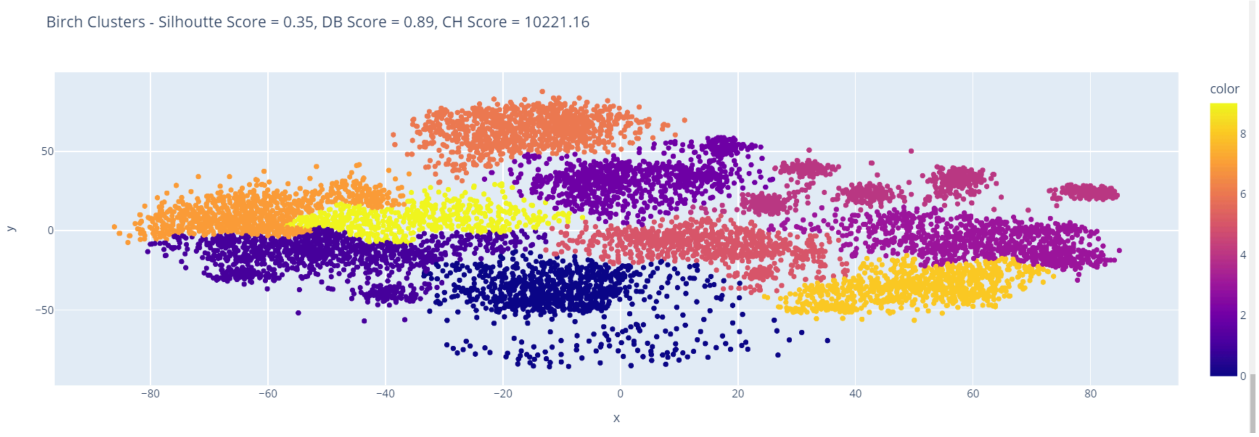
We applied 4 clustering techniques which were K-Means, Agglomerative Clustering, BIRCH and HDBSCAN on the 2D features and compared their performance. The results from K-Means clustering as shown below, depict that linear boundaries between the clusters were made which thereby failed to differentiate between smaller and globular clusters.


KMeans Clusters - Silhoutte Score = 0.39, DB Score = 0.85, CH Score = 12099.75

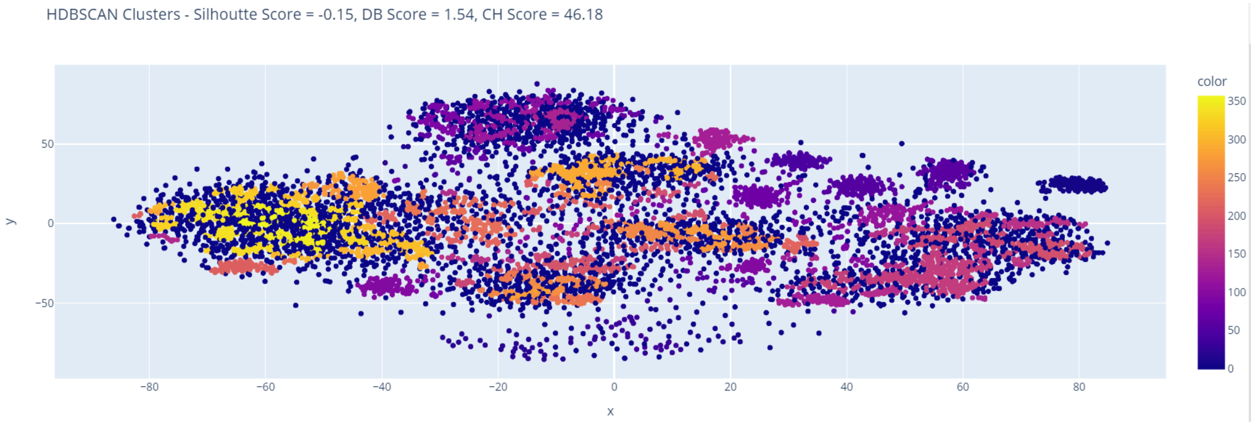
The results from Agglomerative clustering showed that it also wasn't able to assign different clusters to the small purple clusters in top right. It was also dividing one big cluster into two without there being any gap between the two.


Agglomerative Clusters - Silhoutte Score = 0.36, DB Score = 0.81, CH Score = 10388.06

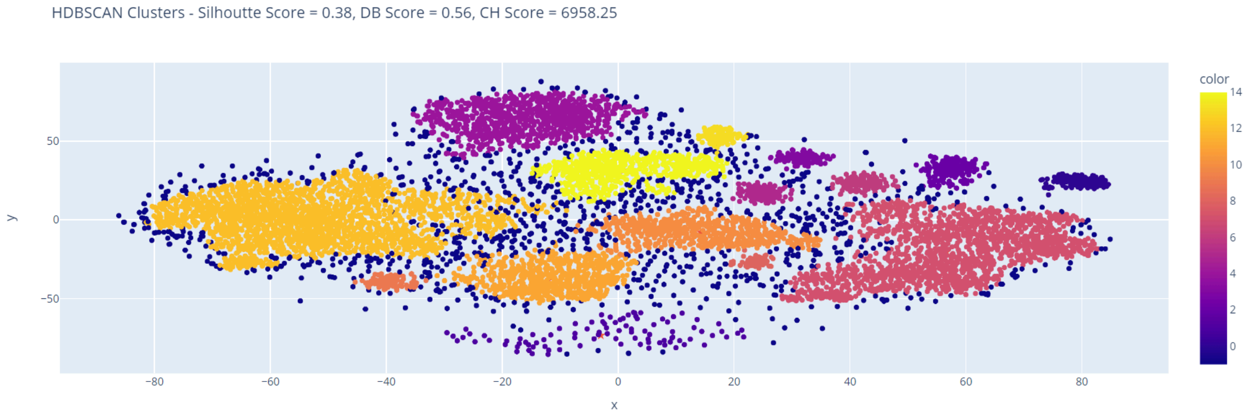
The results from BIRCH clustering faced similar issues as before as can be seen below.


Birch Clusters - Silhoutte Score = 0.35, DB Score = 0.89, CH Score = 10221.16

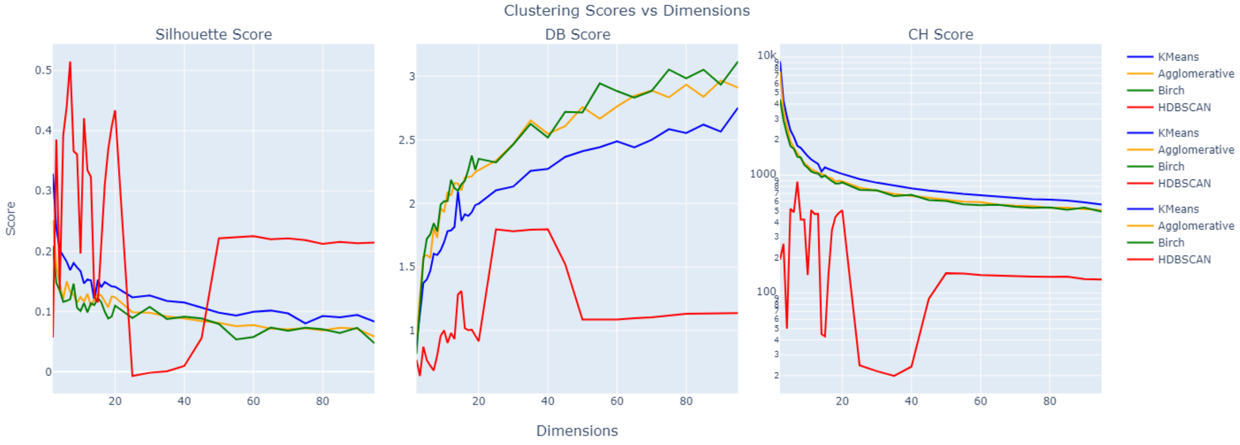
The results from HDBSCAN when looked at first glance were very poor since it created up to 350 clusters with many outliers. Then, we noticed that unlike other algorithms, the number of clusters was not to be pre-assigned and also that it had tunable parameters such as ‘minimum cluster size’ and ‘minimum samples’.



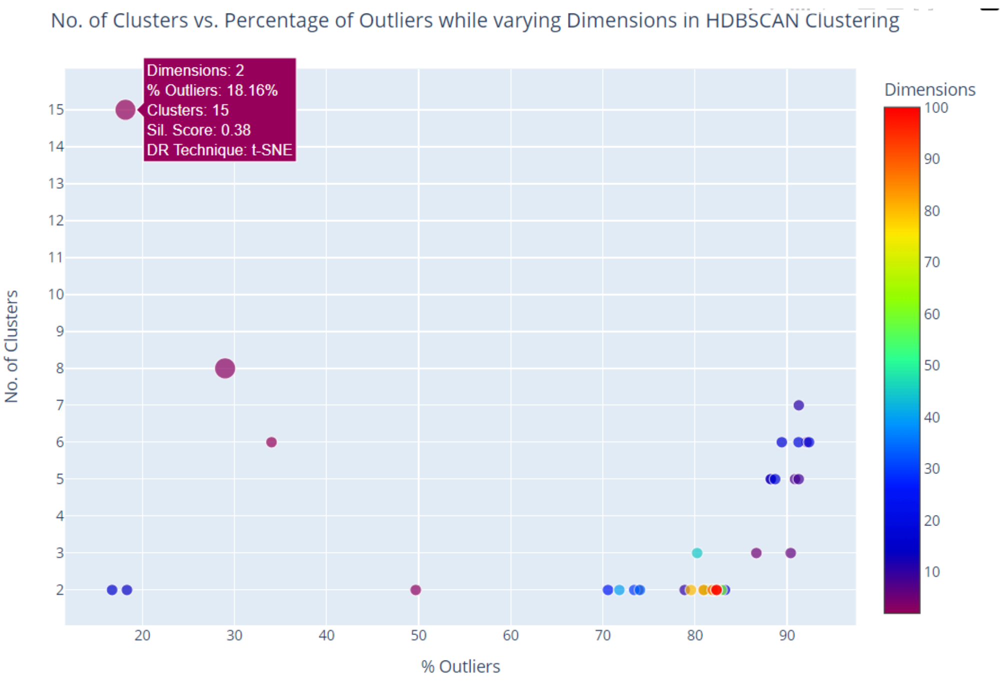
We tuned these parameters and finally at minimum cluster size = 30 and minimum samples = 20, the optimal results were obtained as shown below. It created 15 clusters counting every sub-group of reviews that could be visualized in the graph, with a good enough Silhouette Score at 0.38.



We analyzed how clustering scores obtained using metrics like Silhouette Score, Davies-Bouldin Index and Calinski–Harabasz Index varied, as the number of dimensions of input features and clustering algorithm used were varied. The result of this analysis is shown below. We observe that as the dimension of features increases the values of the scores degrade in all clustering algorithms except HDBSCAN which showed noisy behavior due to the presence of outliers. It got a high silhouette score at 0.5 when dimensions were 7.



To analyze how HDBSCAN performed at different dimensions, we plotted the following graph which shows multiple information about the results. The percentage of outliers had to be minimized and the number of clusters should have been at least 10. We observed that when the Silhouette coefficient was high, the percentage of outliers was also high at 80-90% and the number of clusters was very low at 2-3. The best results from HDBSCAN was obtained only when 2D features from t-SNE were used as input at which 15 clusters were made at only 18% outliers.



e) ANN Classifier

Artificial Neural Network (ANN) classifier tasked with categorizing inputs into 15 distinct classes, each representing one of the cluster centroids derived from our analysis. Initially, we observed significant variance in the support count among these clusters, with some containing a substantial number of data points while others representing minority clusters.

To address this challenge, we implemented the Synthetic Minority Over-sampling Technique (SMOTE). This technique effectively increased and balanced the number of points within minority clusters, mitigating bias and ensuring all clusters contribute meaningfully to the model's training process.

The ANN classifier itself operates as a simple yet robust classifier, processing features extracted from n-grams and preserving their contextual relationships. Despite its straightforward role, it exhibits strong performance, demonstrating state-of-the-art accuracy levels. The classifier's architecture, though deep and complex, effectively distills complex information into actionable insights, making it suitable for real-world applications such as sentiment analysis or customer feedback interpretation.

During implementation, various hyperparameters were tuned, including epochs, optimizers, and activation functions, to optimize the model's performance. Through careful experimentation, we found that the Adam optimizer paired with the Tanh activation function yielded optimal results. Although the model typically required training for more than 30 epochs due to its complexity, the performance gains justified the computational expense.

Results and Findings

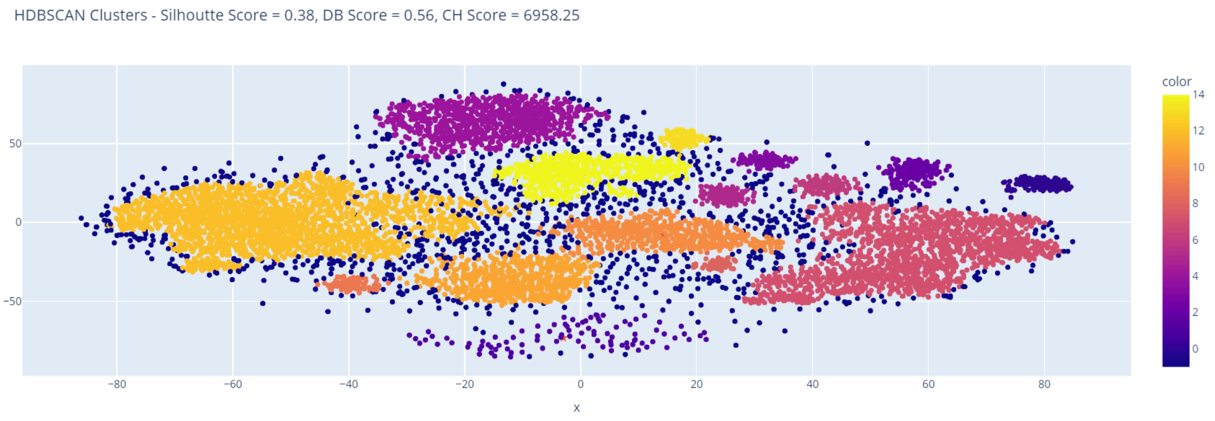
a) Findings from Hyperparameter Tuning

The vigorous process of tuning various hyperparameters such as dimensionality technique used, clustering algorithm used, number of clusters and dimensions of feature matrix, resulted in obtaining the best set of values for these parameters, which are shown in table below:

Hyperparameters	Dimensionality Technique	Dimensions of text features	Clustering Algorithm	No. of Clusters
Best Values	t-SNE	2	HDBSCAN	15

b) HDBSCAN clustering results

The obtained clusters after setting these parameters in our pipeline, is visualized in the graph shown below:



All of the clusters are well separated with minimal number of outliers. Each group of packed reviews however big or small its size, has been assigned as a separate cluster. The number of reviews in each cluster is shown in the table below:

No. of Reviews in Each Cluster obtained from HDBSCAN							
Cluster ID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Count	151	433	220	122	1005	156	156	1820
Cluster ID	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Outliers**
Count	72	115	761	769	2208	117	669	1947

These clusters were evaluated based on the below 3 metrics. Their values are:

1. Silhouette Coefficient = 0.38
2. Davies-Bouldin Index = 0.56
3. Calinski–Harabasz Index = 6958.25

These values also indicate that the clusters obtained are good indeed.

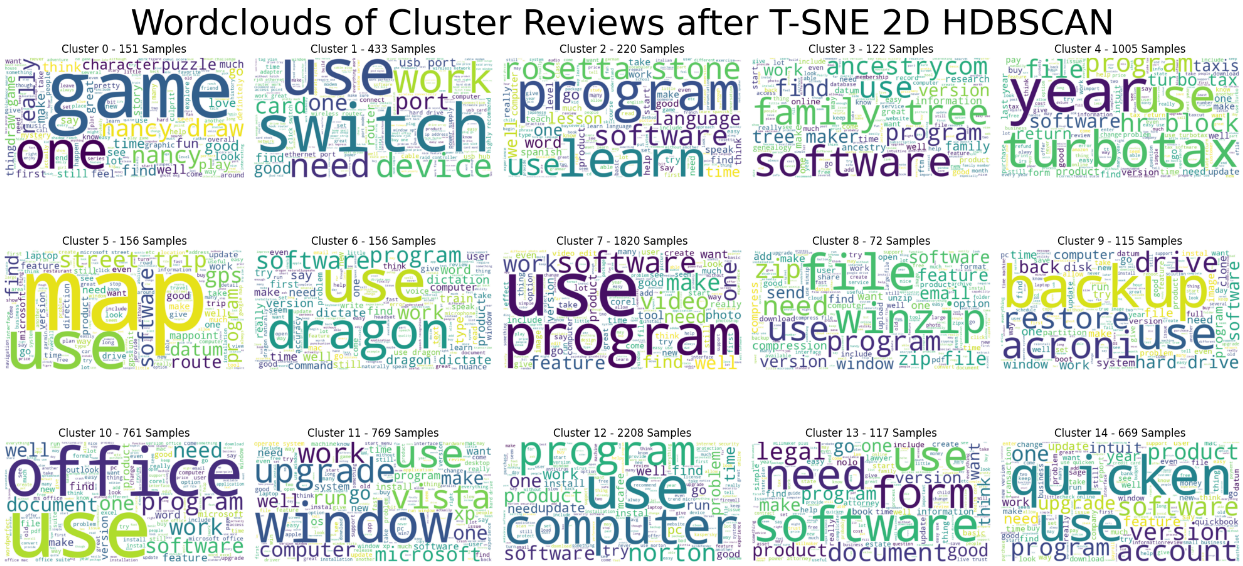
c) Cluster Topic Extraction and Opinion Analysis

For each cluster, their 2D centroids were mapped to their 100D counter-parts. Using the Word2Vec model, the 10 closest words to these 100D cluster centroids were obtained to understand the topics and their opinions, discussed in those reviews. These are as shown in the following table:

Cluster ID	Most Representative Terms	Topic
0	puzzle game nancy creepy spooky storyline twist plot she laugh	Puzzle Game
1	wifi ethernet lan ghz 80211a 80211b wireless device ethernettowifi network	Networking
2	teach language german learn spanish lesson english chinese japanese class	Language Learning

3	ancestrycom family ancestry online genealogy research tree gather cloud software	Family
4	turbotax taxis year return tt tax accountant form turbo peachtree	Tax
5	gps route reroute trip street st map recognition delorme mappoint	Navigation
6	dragon msre dictate command spell sentence word dictation punctuation macspeech	Speech Recognition
7	edit fun novice lot amateur video newbie x3 beginner look	Video Editing
8	unzip zip zipping winzip file self extracte decompress 7zip archive compression	File Compression
9	backup restore reinstall reboot disk acroni boot install restart ati	Disk Backup
10	office doc wp word ms xlsx document docx open wordperfect	MS Office
11	window xp vista desktop 64bit machine run hardware os motherboard	Windows
12	install instal uninstall computer uninstall mcafee antivirus norton virus kaspersky	Anti-Virus
13	form estate legal lawyer attorney willmaker w2s plan deduction paperwork	Legal Paperwork
14	quicken quickbook qb transaction account payee version qb mac exist force	Finance

The most frequently occurring words in each cluster are visually shown in terms of WordCloud plots in which the bigger the words, the more frequently they occur.



The observations from these WordClouds clearly match with the most representative terms extracted using their cluster centroids. The different software products that are talked about are: Nancy Drew, Rosetta Stone, Ancestry.com, TurboTax, MapPoint, Dragon, Corel, Winzip, Acronis, MS Office, Windows, Norton Antivirus, WillMaker and Quicken. Words such as “great, pretty, highly recommended, good, problem, issue, bug, complicated, etc” describe the opinions of the customers towards these software products.

The labels obtained from these clusters were used to train a deep neural network for classification into 15 classes.

d) Results from Deep ANN Classification

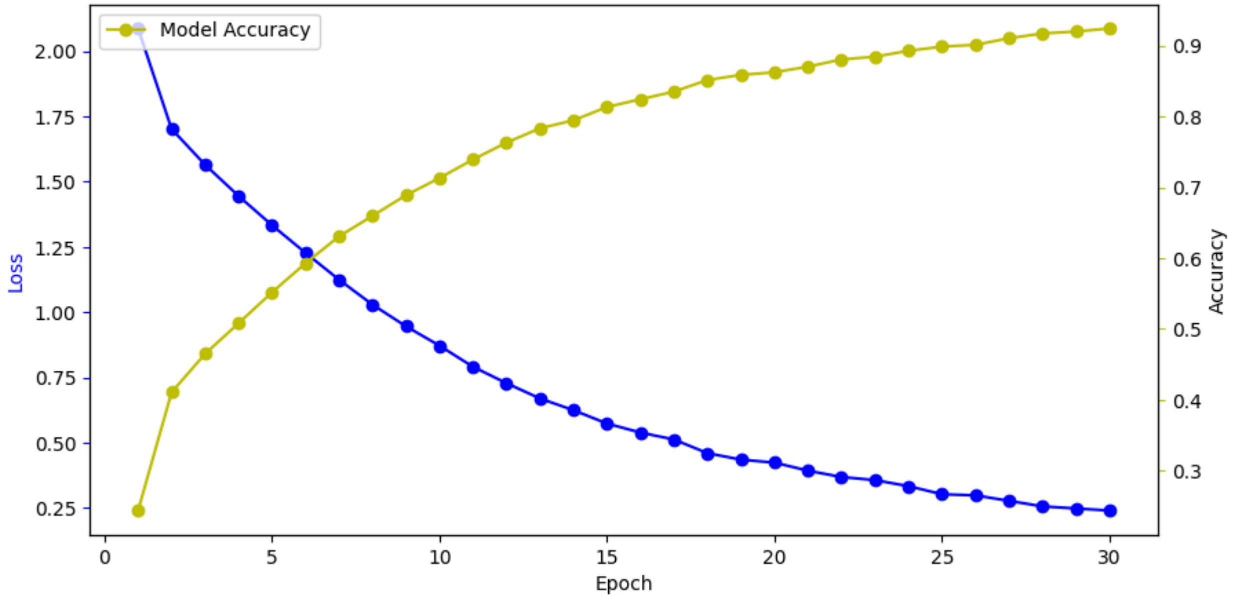
To test our work, we present the application of a deep neural network as a classifier for labeling a dataset into 15 centroid classes derived from cluster analysis. The network processes features extracted from n-grams, preserving their contextual relationships. The deep classifier demonstrates state-of-the-art (SOTA) performance and accuracy. Hyperparameter tuning, including epochs, optimizers, and activation functions, significantly influences model performance.

Optimal performance is achieved with the Adam optimizer and Tanh activation function, typically requiring training for more than 30 epochs due to the network's complexity.

Performance metrics are detailed in the table below, illustrating the impact of various hyperparameters on model effectiveness.

Activation function	Accuracy (macro)	Precision	F1-score	Optimizer	Number of epochs
ReLU	88%	89%	87%	adam	30
Tanh	93%	93%	93%	adam	30

The loss versus accuracy variation across epochs demonstrates the robustness of our model. With an accuracy of 93% and a loss lower than 3%, our classifier exhibits strong performance, indicating its effectiveness in accurately categorizing inputs into the desired classes.



Conclusion and Future Scope

In this comprehensive project report, our focus is on "Topic Extraction and Opinion Analysis of Customer Reviews using Deep Learning." The literature survey delves into six relevant research papers, showcasing cutting-edge approaches in addressing diverse challenges related to customer sentiment analysis. Key issues include depression detection, opinion mining, and identification of customer complaints in the context of cloud service providers. Furthermore, the exploration of online customer touchpoints and the evolution of customer discussions and opinions to shed light on critical areas for enhancing customer experience. Through this literature review, the identified research objectives prioritize topic analysis, sentiment analysis, opinion review, customer emotion detection, and topic extraction and analysis as central to our project's scope.

For the implementation phase, we have meticulously chosen tools and technologies that align with the project's objectives. Obtaining a dataset from UC San Diego, we employ Python, Google Colab, and various libraries such as NLTK, Spacy, Yake and Gensim for Natural Language Processing. The critical task of topic extraction is approached via clustering using advanced non-linear algorithms such as BIRCH, HDBSCAN, ensuring a nuanced understanding of the topics being discussed in the reviews and their opinions about them. These tools and methodologies, integrated into our project, position us to contribute meaningfully to the burgeoning field of sentiment analysis and topic extraction from customer reviews.

The research work introduces a novel approach to customer feedback analysis through customer topic modeling, which preserves and depicts customer opinions and topics discussed via relational clusters. This advancement moves beyond traditional sentiment analysis by extracting vital information about customer opinions and specific product improvements. By clustering reviews into 15 distinct clusters derived from Word2Vec encodings, we obtain labels for training a deep neural network classifier. Throughout our methodology, we emphasize the importance of preprocessing, feature extraction, and clustering techniques to enhance the quality and usability of the data. Notably, we implement the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance within the clusters, ensuring all contribute meaningfully to the model's training process. Our deep neural network classifier, with its intricate architecture and optimal hyperparameter configurations, achieves state-of-the-art performance, showcasing an accuracy of 93% and a loss lower than 3%. These results underscore the efficacy of our approach in accurately categorizing inputs into 15 distinct classes derived from customer feedback analysis. Moving forward, our methodology holds promise for real-world applications such as sentiment analysis and customer feedback interpretation, ultimately enabling businesses to enhance their understanding of customer sentiment and improve overall customer satisfaction.

References

[1] Firoz, Neda & Beresteneva, Olga & Vladimirovich, Aksyonov & Tahsin, Mohammad. (2023). Enhancing Depression Detection through Advanced Text Analysis: Integrating BERT, Autoencoder, and LSTM Models. 10.21203/rs.3.rs-2782391/v1.

[2] Hajek, P., Barushka, A., Munk, M. (2020). Opinion Mining of Consumer Reviews Using Deep Neural Networks with Word-Sentiment Associations. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 583. Springer, Cham.

[3] S. Alghamdi, "Toward Identifying Customer Complaints of Cloud Service Providers Using Topic Modeling and Sentiment Analysis," 2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS), Abu Dhabi, United Arab Emirates, 2023, pp. 1-6, doi: 10.1109/SNAMS60348.2023.10375466.

[4] Z. Yilin, A. Fayoumi and A. Shahgholian, "Understanding Online Customer Touchpoints: A Deep Learning Approach to Enhancing Customer Experience in Digital Retail," 2023 9th International Conference on Information Technology Trends (ITT), Dubai, United Arab Emirates, 2023, pp. 193-198, doi: 10.1109/ITT59889.2023.10184269.

[5] S. F. Eletter, K. I. AlQeisi and G. A. Elrefae, "The Use of Topic Modeling in Mining Customers’ Reviews," 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 2021, pp. 1-4, doi: 10.1109/ACIT53391.2021.9677049.

[6] D. Suryadi, H. Fransiscus and Y. G. Chandra, "Analysis of Topic and Sentiment Trends in Customer Reviews Before and After Covid-19 Pandemic," 2022 International Visualization, Informatics and Technology Conference (IVIT), Kuala Lumpur, Malaysia, 2022, pp. 172-178, doi: 10.1109/IVIT55443.2022.10033397.