**ANL588 FINAL REPORT**

**USING SENTIMENT AND THEMATIC ANALYSIS TO EXPLORE KEY BUSINESS ISSUES IN LUSH SINGAPORE**

**Logo, company name

Description automatically generated**

**Submitted by**

**ATIQAH ZAKIRAH BINTE ZULKIPLI**

**SCHOOL OF BUSINESS**

**Singapore University of**

**Social Sciences**

**Presented to Singapore University of Social**

**Sciences in partial fulfillment of the**

**requirements for the**

**Degree of Graduate Diploma**

**in Analytics and Visualisation**

**2022**

**ABSTRACT**

Over the last 10 years, Lush Singapore has been building itself as a high quality and well-reputable brand that we know of today. However, the company still faces major business challenges that has been affecting its business performance. This presents an opportunity to explore and understand the key issues affecting Lush Singapore through the use of automation and machine learning. A sentiment analysis model was created to classify customer product reviews into positive or negative sentiment polarity. Three supervised machine learning algorithms were compared against each other in respect to their performance. RAKE (Rapid Automatic Keyword Extraction) algorithm was applied to customer product reviews to determine its effectiveness for a small-scale thematic analysis, which aims to uncover positive and negative factors affecting Lush Singapore’s business performance. Results show that Logistic Regression is the best performing model with an accuracy of 83.64%. Recommendations were offered on how to fully utilize RAKE algorithm to produce a more comprehensive thematic analysis. Overall, it was observed that the combination of a sentiment analysis model along with thematic analysis has the potential to tackle Lush Singapore’s negative business issues.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Abstract |  | i |
| Chapter One | Introduction | 1 |
| Chapter Two | Literature Review | 3 |
| Chapter Three | Data | 5 |
| Chapter Four | Sentiment Analysis Modelling and Evaluation | 9 |
| Chapter Five | Thematic Analysis | 20 |
| Chapter Six | Conclusion and Future Work | 24 |
| References |  | 26 |
| Appendix |  | 28 |

**Introduction**

For businesses to continue growing and improving it is important that they consider the various factors that could affect their potential for success. A crucial factor is customer needs satisfaction. The success of a business largely depends on whether they are able to meet their customers’ needs and how satisfied their customers are with their service or product. There have been many techniques and approaches to better understand customers from market surveys to the use data analytics. Due to the advancement of technology, there have been more efficient and optimized ways to gather customer information through the use of artificial intelligence. Such artificial intelligence includes technologies such as computer vision, language processing, and machine learning, which are applied to businesses processes to uncover business insights in the effort to improve business performance.

Lush Fresh Handmade Cosmetics is a British cosmetics retailer which produces vegetarian and vegan cosmetics for the face, hair and body. The Lush brand revolves around its core brand values which focuses on being environmentally sustainable, hand-made, against animal testing, and sourcing fair-trade and ethical ingredients. Lush Singapore first opened its doors in 2012 with its first outlet being at Wisma Atria. Over the past ten years Lush Singapore has been building its brand and reputation as a welcoming and inclusive store. The most important role as an on-floor employee at Lush is the engagement with customers. Lush Singapore focuses heavily on building rapport with customers and engaging in employee-led demos to showcase the products in an effort to produce sales. However, there are other factors which play an equally important role such as product quality and customer relationship management.

Lush uses fresh ingredients in their cosmetics which means that the products have a shelf life and expiry date. All products have an “off-shelf” date (which is typically seven months prior to the expiry date) when it can no longer be sold to customers. This is to ensure a certain level of freshness during the time the customer purchases the item as well as to ensure the customer has a decent amount of time to use and keep the product before it expires. Often times products meet their off-shelf date before they are sold. On average, there are about two shopping carts worth of products that go off-shelf at the end of every month which results in a significant amount of product wastage. Lush Singapore stores are expected to meet their monthly target sales amount to offset the cost of these off-shelf products, however stores are not meeting their target sales amount.

By applying sentiment and thematic analysis, it can provide insights into why products are becoming off-shelf instead of being sold. It can uncover areas of improvement in product quality and a solution to product wastage. Like other retail companies, Lush receives both negative and positive feedback on the regular. This feedback comes in the form of a written email or customer product review often times indicating an issue towards a product or service. Sentiment analysis can be utilized to gauge the urgency of the message. An urgent message typically has a negative sentiment. By using sentiment analysis, reviews and messages with a negative sentiment can be prioritized to ensure it gets resolved as soon as possible, which increases response rate. There are different ways in which sentiment and thematic analysis can be applied to a business. It provides context to a business problem and potentially a solution to solve it.

**Literature Review**

Thematic analysis is a method of analysing qualitative data. Qualitative data is non-numeric information such as interview transcripts, open-ended survey questions, audio-visual recordings and images. Thematic analysis entails searching across such qualitative data to identify, analyse, and report repeated patterns in an effort to understand a set of experiences, thoughts or behaviours (Medelyan, 2018). Over the years, there has been many applications of thematic analysis in business intelligence and business decision-making. Often times sentiment and thematic analysis go hand in hand to uncover new business insights, with sentiment analysis being the first step followed by thematic analysis. Sentiment analysis is a form of text analytics that uses natural language processing (NLP) and machine learning to determine the sentiment of a text which can be positive, neutral, or negative. Unlike sentiment analysis, thematic analysis is more actionable. Knowing whether a product review is positive or negative is a great classifier, but it does provide an answer as to why it is positive or negative. Thematic analysis uncovers themes which allows businesses to accurately target areas of improvement.

There have been many studies applying sentiment and thematic analysis to real-world business problems. Olagunju et al. (2020) conducted a study exploring the key issues affecting mobile e-commerce through the use sentiment and thematic analysis and discovered that the main business issues were bad customer service, high shipping fees, and poor packaging, sudden cancellation without notice, delivery mistakes, purchased tracking issues, insufficient product information, and fake sellers. Uncovering these business issues provides e-commerce businesses with the information they need to implement the appropriate measures to improve their business. Thematic analysis can also be utilized to determine the effectiveness of a product. Oyebode et al. (2020a) conducted a study evaluating mental health mobile applications based on its user reviews using a combination of sentiment and thematic analysis. It was discovered that the factors negatively affecting the effectiveness of mental health apps were content issues such as low-quality content, amateur delivery, inadequate free content, limited content, and counterproductivity.

Thematic analysis can also be applied on a larger scale such as influencing policy making and understanding societal behaviour. During the COVID-19 lock down, there was a massive change in societal behaviour when physical distancing policies were implemented, and the main form of communication went from physical to online. Another significant outcome of the lock down was the personal opinions regarding the pandemic being posted on social media. This provided the data to conduct analysis on public perceptions of their government and health agencies. Oyebode et al. (2020b) conducted a study on the health, psychological and social issues emanating from COVID-19 using data from social media through the use of text mining and thematic analysis. From the study, various negative themes were discovered ranging from health-related issues such increased morality and struggling health systems, as well as psychological issues such as frustration due to life disruption and expression of fear. By identifying negative themes surrounding the pandemic, interventions were recommended to help government, health professionals and institutions, and individuals in their effort to curb the spread of COVID-19 as well as minimize its impact on society. From these studies, it is clear that sentiment and thematic analysis can be applied across various industries and real-world problems, with its results having the potential to enact significant change and improvement.

**Data Source**

Sentiment and thematic analyses were conducted to further understand customer brand perception and customer product experience for Lush Singapore. These analyses were conducted using Lush customer product reviews from the United States. Due to a lack of Singaporean reviews, data was sourced from the Lush USA website instead. Compared to other countries, Lush USA has the highest amount of customer product reviews allowing for a large and comprehensive dataset. The insights discovered from analyses using USA data can be rightfully applied to a Singapore context as both countries share the same products and formulas. These product similarities preserve the quality and accuracy of the data and do not compromise the overall analysis. Table 1 shows the URLs of the web pages where reviews were sourced from.

|  |  |  |
| --- | --- | --- |
| Dataset | Number of rows | URL |
| new | 973 | https://www.lushusa.com/hair/shampoo-bars/new/9999905128.html |
| coco\_rice | 126 | https://www.lushusa.com/hair/shampoo-bars/coconut-rice-cake/05315.html |
| soak\_float | 370 | https://www.lushusa.com/hair/shampoo-bars/soak-and-float/05130.html |
| jason\_argan | 735 | https://www.lushusa.com/hair/shampoo-bars/jason-and-the-argan-oil/05125.html |

Table 1. Data source table listing dataset, number of rows, and URL source

**Data Collection using Web Scraping**

Data was collected using a Python web scraper that was built using a combination of different Python libraries and Javascript rendering service. These include BeautifulSoup and Pandas libraries, and Splash rendering service. Web scraping is an automated method of extracting large amounts of data from websites. Websites typically contain unstructured data which is then stored in a structured format in the process of web scraping. BeautifulSoup is a Python package for parsing HTML and XML documents. It creates parse trees for parsed pages that can be used to extract data from HTML. Pandas is a powerful Python library used for data manipulation and analysis. Its main function is preparing the data for pre-processing by storing scraped data into excel format using *.to\_excel()* method.[[1]](#footnote-1) Splash is a Javascript rendering service which is run as a container service on Docker, which is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers[[2]](#footnote-2). This service is accessed using local host which then allows scraping web data from the respective URLs.

It is important to collect the necessary data for analysis. This includes the review title, full text, and rating. This was achieved by identifying the respective class and tag names. Web scraping product reviews from multiple pages uses the process of pagination which includes determining page URL and scraping data from the respective URL. However, Lush USA uses dynamic webpages where the URL remains unchanged. Instead of relying on the URL, the web scraper continues scraping data as long as a “Next” button exists. This was achieved by determining the class name of the HTML element “Next” button and retrieving the respective page number and HREF. Once all data has been scraped, it is stored in an excel format for further processing. Table 2 below lists out the data attributes and its descriptions before data pre-processing.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of attributes | | 3 | |
| Attribute name | Type | | Description |
| title | String | | Product review header |
| rating | Integer | | Product review rating on a scale of 1-5 |
| body | String | | Full product review |

Table 2. Data attributes before pre-processing

**Data Pre-processing**

Before the data can be used for machine modelling, it has to undergo text cleaning and data wrangling. Data wrangling is also required to combine multiple datasets as well as assign meaning to the text data. To conduct meaningful sentiment analysis, the machine learning model requires a balanced dataset, which has an equal amount of positive and negative data. While acquiring data, it was observed that there was an abundance of positive reviews. A significant amount of positive data had to be removed from the dataset to match the lesser number of negative data to achieve a balanced dataset. This meant removing a portion of product reviews with a rating of 4 and above. After data wrangling has been completed, there was a total of 774 rows (387 positive reviews and 387 negative reviews) of data. The text data was then classified into positive and negative sentiment by assigning a sentiment score to each review according to rating. For reviews with a rating of 4 and above, a sentiment score of 1 was assigned. Reviews with a rating of 3 and below are assigned a sentiment score of 0. These value are stored in a new column named “sentiment”.

Text cleaning is crucial as it removes unwanted characters from the text that would hinder machine learning analysis. NLP techniques are applied to text data to prepare data for the machine learning model. These NLP functions are available in Python NLTK (Natural Language Tool-Kit) library. It includes text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning (NLTK 2019). The process of cleaning text data was to remove stop words which are a set of commonly used words. In the English language, it includes words such as “the”, “is”, and “and”, followed by removing non-alphabetic characters which include emoticons, punctuation and exclamation marks, and numbers. Text data was then stemmed using a process called stemming which reduces a word to its stem word. Snowball stemmer was chosen for this process as it is the fastest and most logical stemmer as compared to other stemmers such as the Lancaster stemmer and Porter stemmer (Heidenreich 2018). Stemmers such as Lancaster stemmer has a tendency of excessively trimming words which would render the stems non-linguistic, meaningless and unusable for analysis. Since sentiment analysis only requires the full review text and sentiment, title was removed from the final usable dataset. Table 3 lists all the attributes of the final dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of attributes | | 2 | |
| Attribute name | Type | | Description |
| review | String | | Full review text after NLP text cleaning |
| sentiment | Integer | | Either 1 or 0  0 – Negative sentiment  1– Positive Sentiment |

Table 3. Data attributes of dataset used for modelling

**Sentiment Analysis Model**

Scikit-learn was heavily utilized in creating a supervised machine learning model for sentiment analysis. This sentiment analysis model aims to classify a review with a positive or negative sentiment polarity based on how well it is able to learn and train from the given dataset. Scikit-learn is a machine learning library for Python. It features various classification, regression, and clustering algorithms. Scikit-learn interoperates with other Pythonic numerical and scientific libraries NumPy and SciPy (Wikipedia Contributors 2019)[[3]](#footnote-3). Its feature extraction and classification algorithm features are applied to the data to create different supervised machine learning models, which are then compared to each other to determine the most effective one for sentiment analysis. Figure 1 is an overview of the modelling process. Each stage will be elaborated further in detail.

Figure 1. Modelling process

**Feature Extraction with TF-IDF**

Feature extraction is the initial step in machine learning modelling whereby it derives values also known as features from a set of measured data. These features are intended to be informative and non-redundant to facilitate the subsequent learning and generalization steps to better interpret the data. Feature extraction reduces the number of resources required to describe a large dataset. Often times, problems arise from analysing a large dataset as it can cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction prevents this by constructing combinations of variables (Wikipedia Contributors 2021)[[4]](#footnote-4).

This stage in the modelling process applies TF-IDF algorithm on text data “review”. Term Frequency Inverse Document Frequency (TF-IDF) is an algorithm to transform text into a meaningful representation of numbers which is used to fit machine learning algorithms for prediction (Stecanella 2019). Features are extracted by using TF-IDF vectorizer function *Tfidvectorizer* imported from Scikit-learn’s *sklearn.feature\_extraction.text* class. TF-IDF vectorizer calculates how relevant a word in a corpus is to a text by multiplying term frequency and inverse document frequency. It converts data attribute “review” to a matrix of features in which the number of columns represents the number of features. TF-IDF vectorizer is also a weighting system that assigns a score to each word in a document. A high score indicates a high weightage which means that the term is a rare term and has a high relevance in the body of text. A low score indicates the opposite. TF-IDF vectorizer also has a built-in tokenizing function which splits paragraphs into smaller units that can be more easily assigned meaning. These units can be individual words or terms. After applying TF-IDF, there are a total of 1884 features. These features are used as labels in the subsequent stages of training and testing the machine learning model.

**Create and Train Model**

The extracted features and data attribute “sentiment” are used to conduct modelling and training, where the extracted features are the labels, and the sentiment is the target. The data was split into a training and testing set. Various classification algorithms are applied to the training set. These algorithms are generally used in supervised machine learning, which uses labelled datasets to train algorithms to classify and predict the outcomes accurately. It relies on labelled input data to learn a function that produces an appropriate output when given new unlabelled data.[[5]](#footnote-5) The classification algorithms used for modelling are Naïve Bayes Classifier, Logistic Regression, and K-Nearest Neighbour. As observed in the literature review, these classification algorithms have a high F1-score also known as the model accuracy score. It is well-reasoned to believe that these algorithms are effective in creating an accurate sentiment analysis model.

To create the training and testing sets, *train\_test\_split* function was utilized. This function was imported from Scikit-learn’s *sklearn.model\_selection* class. 80% of the data was assigned to a training set, while the remaining 20% was assigned to a test set. This will result in the variables X\_train, y\_train (training set) and X\_test, y\_test (testing set) as shown in Table 4.

|  |  |  |
| --- | --- | --- |
| Variable | Dataset | Number of rows |
| X\_train | Training | 619 |
| y\_train |
| X\_test | Testing | 155 |
| y\_test |

Table 4. Datasets after splitting data using train-test-split method

The algorithms are then applied to the training set through the use of Scikit-learn’s algorithm classes which include *sklearn.naive\_bayes*, *sklearn.linear\_model*, *sklearn.neighbours*, creating three different models: Naïve Bayes Classifier model, Logistic Regression model, and K-Nearest Neighbour model.

**Naïve Bayes Classifier Model**

Naïve Bayes Classifier is a supervised learning algorithm which is based on Bayes Theorem and is used for solving classification problems, mainly text classification. It is a simple and effective classification algorithm which helps in building fast machine learning models that make quick predictions. One of the main applications of Naïve Bayes Classifier is sentiment analysis. There are three types of Naïve Bayes models: gaussian, multinomial, and Bernoulli. The sentiment analysis model will utilize Multinomial Naïve Bayes (MNB) which is a classifier used when the data is distributed nominally. It is primarily used in document classification problems, whereby the document belongs to a specific category (Javatpoint Contributors n.d.)[[6]](#footnote-6). Olagunju et al. (2020) compared the F1-score of five different supervised machine learning models which were Stochastic Gradient Descent, Logistic Regression, Support Vector Machine, Multinomial Naïve Bayes (MNB), and Random Forest, and found that MNB was the best classifier with an F1-score of approximately 82%. An F1-score is an important evaluation metrics in machine learning. It sums up the predictive performance of a model by combining the precision and recall metrics.

**Logistic Regression Model**

Logistic Regression (LR) is another popular supervised machine learning algorithm. It is used for predicting the categorical dependent variable using a given set of independent variables. Unlike its familiar term Linear Regression, which is used for solving regression problems, LR is used for solving classification problems, specifically by calculating or predicting the probability of a binary event occurring. LR uses the Sigmoid Function which maps any real values into another value between 0 and 1 (Javatpoint Contributors n.d.)[[7]](#footnote-7). It’s application in machine learning maps predictions to probabilities. Similar to MNB, LR has been used extensively in research studies. Oyebode et al. (2020a) utilized various different super machine learning algorithms which included: LR, MNB, and Support Vector Machine, and Stochastic Gradient Descent to conduct sentiment analysis of mental health app user reviews and found that LR has the second highest F1-score of 89.37%.

**K-Nearest Neighbour Model**

K-Nearest Neighbour (KNN) algorithm is a supervised learning classifier which uses proximity to make classifications or predictions about the grouping of an individual data point. It is a versatile algorithm in such that it can be applied to both regression and classification problems. However, its main application is on classification problems basing on the logical assumption that similar points can be found near one another (Javatpoint Contributors n.d.)[[8]](#footnote-8). KNN algorithm is typically implemented in sentiment analysis. An example is the implementation of KNN algorithm for public sentiment analysis of online learning using twitter data. It was observed that the use of KNN algorithm produced an F1-score of 87% (Isnain 2021).

**Model Evaluation**

The models are evaluated on their accuracy scores and Area Under the Receiver Operating Curve (ROC AUC). Once MNB, LR, and KNN models are created, the training set is fit onto the models using Scikit-learn’s *.fit()* method which finds the coefficients for the equation specified via the algorithm being used. After the models have been fitted, *.predict()* method is called to classify incoming data points from the testing set and make predictions. The accuracy score is computed using *accuracy\_score* function from Scikit-learn’s *sklearn.metrics* class. This function computes subset accuracy whereby the set of labels predicted from *.predict()* method must exactly match the corresponding set of labels in the testing set also known as y\_test. AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The ROC AUC is computed using *roc\_auc\_score* function from Scikit-learn’s *sklearn.metrics* class. Table 5 lists the accuracy and ROC AUC scores of all three models.

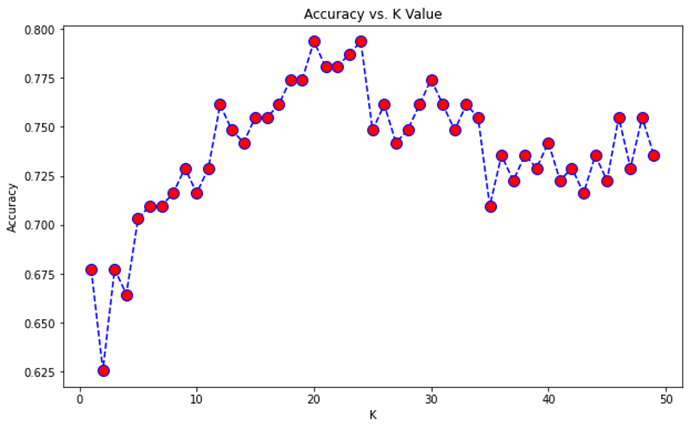
Before evaluating the accuracy of KNN model, the model requires hyperparameter tuning. Finding the optimal value of hyperparameter K is crucial to achieve optimal accuracy. In Scikit-learn, the default value of K is 5. However, this value may not be suitable for this sentiment analysis. Since there are no pre-defined statistical methods to find the most favourable value of K, finding the right value includes trialling and testing different values of K and then choosing the value that has with the highest accuracy score (Band 2020). This was achieved through the use of a for loop which loops a K value of 1 to 50 and then plotting its respective accuracy on a graph as seen in Figure 2. It was observed that K with the value of 19 had the highest accuracy score 79.35%.

Figure 2. Graph plotting accuracy value against k-value

|  |  |  |
| --- | --- | --- |
| Model | Accuracy score (%) | ROC AUC score (%) |
| Multinomial Naïve Bayes | 96.28 | 81.29 |
| Logistic Regression | 96.12 | 83.87 |
| K-Nearest Neighbour | 79.35 | 71.61 |

Table 5. Model accuracy scores and ROC AUC scores after evaluation

**K-Fold Cross Validation**

After evaluating all three models, it was observed that Logistic Regression model performed the best in terms of accuracy and ROC AUC score. Multinomial Naïve Bayes model did just as well in accuracy, but the LR model outperformed it in terms of ROC AUC score. K-Nearest Neighbour model scored the lowest in both metrics. It was observed that the accuracy score changes after each time the model is retrained. It is clear that the models needed to be validated further to be able to come to a well-informed decision as to which model is the most optimal. K-fold cross validation was conducted to achieve a more comprehensive analysis as it generally results in a less biased estimate of the model skill than train-test split which was the method used in the earlier stage to achieve the current accuracy and ROC AUC scores. This validation method is also used to flag problems like overfitting and selection bias and give a better generalized model (Mujtaba 2020).

The method of k-fold cross validation includes splitting the data into K groups. For this analysis, K is set to 10 as the value of 10 has been found through experimentation to generally result in a model skill estimate with a low bias (James et al. 2013). For each unique group, that group is used as a test set while the remaining 9 groups are used as a training set. The model is then fitted onto the training set and evaluated on the test set. Once the accuracy and ROC AUC scores are calculated, the model is discarded. This process repeats itself k times ensuring that each sample is given the opportunity to be used in the test set one time and used to train the model k-1 times. Once the validation process is completed, the skill of the model is based on the mean of the total accuracy and ROC AUC scores.

K-fold cross validation was achieved by first creating a k-fold object by importing *KFold* from Scikit-learn’s *sklearn.model\_selection* class. The dataset containing “reviews” and “sentiment” was split into 10 folds by specifying *n\_splits=10* and using the *split()* method. A for loop then loops through each fold. Within the loop, the training and testing sets are defined, feature extraction is conducted using TF-IDF and the training sets are fitted onto the model. Scores are calculated using *accuracy\_score* and *roc\_auc\_score* functions. The following images in Table 6 shows the scores of MNB, LR, and KNN models. Table 7 lists the overall model performance scores. It was observed that Logistic Regression Model has the highest accuracy and ROC AUC scores of 82.17% and 90.23%, making it the most optimal model for sentiment analysis.

|  |  |
| --- | --- |
| Model | Accuracy and ROC AUC scores |
| Multinomial Naïve Bayes Classifier | Text  Description automatically generated |
| Logistic Regression | Text  Description automatically generated |
| K-Nearest Neighbour | Text  Description automatically generated |

Table 6. Model accuracy scores and ROC AUC scores for each fold

|  |  |  |
| --- | --- | --- |
| Model | mean(Accuracy) % | mean(ROC AUC) % |
| Multinomial Naïve Bayes Classifier | 82.4 | 90.90 |
| Logistic Regression | 82.17 | 90.23 |
| K-Nearest Neighbour | 79.32 | 87.31 |

Table 7. Mean accuracy scores and mean ROC AUC scores representing overall model skill

**Hyperparameter Tuning**

When comparing scores from train-test split method and k-fold cross validation, it was observed that there was a decrease in model accuracy. LR model’s accuracy decreased from 96.12% to 83.73%. According to k-fold cross validation, the latter scores are a more realistic and accurate representation of the model’s skills. However, these scores can be further improved by performing hyperparameter tuning. Unlike parameters, hyperparameters are internal coefficients for a model found by the algorithm and are explicitly specified by the developer during configuration. Hyperparameter tuning includes objectively searching different values for model hyperparameters and choosing a subset that results in a model that achieves the best performance. A common method of hyperparameter tuning is called a grid search, which defines a search space as a grid of hyperparameter values and evaluates every position in the grid (Brownlee 2020).[[9]](#footnote-9) This method is applied onto the current LR model to determine the best hyperparameters for an optimized model.

Not all hyperparameters are equally important. Only some hyperparameters have an outsized effect on the behaviour and performance of a machine learning algorithm, such as the hyperparameters solver, penalty, and C (Brownlee 2019). These hyperparameters are tuned for LR model optimization. Solver is the algorithm used in the optimization problem. Penalty (or regularization) reduces model generalization error and regulate overfitting. Some penalties do not work with some solvers. Lastly, C is a value that controls regularization strength. It works with the penalty to regulate overfitting. Smaller values specify stronger regularization while higher value gives weight to the training data (Gusarov, 2022)[[10]](#footnote-10). Table 8 lists out the hyperparameters and the respective arguments to tune.

|  |  |
| --- | --- |
| Hyperparameter | |
| solver | [‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’] |
| penalty | [‘none’, ‘l1’, ‘l2’, ‘elasticnet’] |
| C | [100, 10, 1.0, 0.1, 0.01] |

Table 8. Lists of hyperparameter arguments

To conduct hyperparameter tuning, the model and hyperparameters are defined with values given in Table 7. Since the grid search method is used, a search space needs to be defined. It is defined as a dictionary using *dict()* function where the names are the hyperparameter arguments. The grid search is conducted using Scikit-learn’s *GridSearchCV* from *sklern.model\_selection* class. It implements a *fit()* and *score()* method to fit the dataset which consists of label data (TF-IDF features) and target data (sentiment) onto the model with the set parameters and return the accuracy score. The results of the grid search are shown in Figure 3. Some combinations were omitted to cut back on warning and errors due to the fact that some penalties do not work with some solvers as mentioned earlier. It was observed that the best accuracy score of 83.64% was achieved when LR model hyperparameter are tuned to *C=1.0*, *penalty=’l2’*, and *solver=’liblinear’*. The LR model accuracy improved by 1.47%, increasing to 83.64% from 82.17% before hyperparameter tuning.

Text

Description automatically generatedFigure 3. Model accuracy scores of different hyperparameter combinations

**Thematic Analysis**

Unlike building a sentiment analysis model, thematic analysis is conducted on a much smaller scale and uses a smaller dataset. There is many thematic analysis software found on the market today such as Thematic, NVivo, and IBM Analytics. However, since these tools are not open access, Python Rake-NLTK library is utilized for this small-scale thematic analysis, specifically it’s RAKE algorithm. The process of thematic analysis involves applying RAKE algorithm to both positive and negative sentiment reviews to identify positive and negative keywords. Thematic analysis can be considered as an optional or additional step to sentiment analysis.

**Application of RAKE Algorithm**

RAKE also known as Rapid Automatic Keyword Extraction is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analysing the frequency of word appearance and its co-occurrence with other words in the text. It is based on the observation that keywords frequently contain multiple words but rarely contain punctuation, stop words, or other words with minimum lexical meaning (Sharma n.d.). Its features contain configurable word and sentence tokenizers and configurable ranking metric. It assigns a score to the keywords retrieved, and the higher the score, the higher importance of the word. It is assumed that the main keywords of the text are considered as the theme of the text.

The dataset consists of only 400 rows, 200 rows of positive sentiment reviews and 200 rows of negative sentiment reviews. Rakewas imported from the rake-nltk package, and Rake object was created using *Rake()* function. To obtain the keywords from the text, *extract\_keywords\_from\_text()* function was applied to the text. During this process, Rake assigns a score to the keywords retrieved. To obtain keywords along with its scores, *get\_ranked\_phrases\_with\_scores()* function was applied to the Rake object. A for loop was used to loop this process for all 400 reviews. RAKE output from 3 different positive and negative reviews and their respective themes are shown in Table 9 and 10, representing a small sample output from the thematic analysis conducted.

|  |  |  |
| --- | --- | --- |
| Full review | RAKE output | Themes (top 3) |
| I had a really stressful time in my life a year and a half ago, and was losing so much hair. Every time I shampooed I would get so nervous because so much hair would be in the drain. I went for the New bar as a last ditch effort to keep my scalp clean from bad shampoos and give it all the love I could. I'm on my second new bar, my hair is growing back like nuts!! So happy I made the switch to the new bar, and I'll keep using lush bars in the future now that I don't need plastic bottles anymore! | [(25.0, 'growing back like nuts !!'), (16.0, 'need plastic bottles anymore'), (14.5, 'keep using lush bars'), (9.0, 'last ditch effort'), (8.5, 'really stressful time'), (7.666666666666668, 'second new bar'), (7.0, 'much hair would')] | Growing, plastic bottles, keep using |
| I have loved all of the shampoo bars that I've used and this ones the same. I have been using the same bar for  Months and still have a bunch more product to use. The scent smells great - cinnamon! It does not have a "tingling" effect though, and I haven't noticed much difference in the length but otherwise it cleans and lathers nicely. | [(9.0, 'scent smells great'), (9.0, 'noticed much difference'), (4.0, 'shampoo bars'), (4.0, 'lathers nicely'), (4.0, 'effect though'), (1.0, 'using'), (1.0, 'used'), (1.0, 'use'), (1.0, 'tingling'), (1.0, 'still'), (1.0, 'product'), (1.0, 'otherwise'), (1.0, 'ones'), (1.0, 'months'), (1.0, 'loved'), (1.0, 'length'), (1.0, 'cleans'), (1.0, 'cinnamon'), (1.0, 'bunch'), (1.0, 'bar')] | Smells great, noticed difference, lathers nicely |
| I was looking for something that would make my hair grow longer faster, and a friend who works at Lush recommended this product. After two washes with it, I already feel improvement in my hair quality and length. Awesome shampoo. Works miracles. | [(15.0, 'hair grow longer faster'), (9.0, 'already feel improvement'), (5.0, 'hair quality')] | Hair grows, improvement, hair quality |

Table 9. Full text of positive review, RAKE output, and positive themes

|  |  |  |
| --- | --- | --- |
| Full review | RAKE output | Themes (top 3) |
| This is my fourth different solid shampoo, this one has been the worst so far... I love the scent and the description of what it's supposed to do. However this bar gave me an extreme dandruff and almost psoriasis like reaction on my scalp. The front middle section of my scalp had flakes the size of large rice I had to pull out and my scalp was red and raw underneath and sore. I've never had this happen! | [(15.5, 'fourth different solid shampoo'), (15.0, 'almost psoriasis like reaction'), (9.0, 'good shampoo bar'), (9.0, 'front middle section'), (9.0, 'always fall short'), (5.0, 'sensitivity reaction') | Psoriasis, sensitivity, different solid shampoo |
| I'm a big fan of shampoo bars, but not a fan of "new." For starters, I've never had problems with dandruff, my scalp isn't too dry, but when I used this bar my scalp turned into an irritated flaky mess. I was wearing headscarves for a couple weeks to cover up the skin flakes that were very, very visible. The only other problem was the smell, I'm a fan of cinnamon but this was a bit strong in my opinion. Other than that, it was very easy to use and made my showering process a lot faster | [(9.0, 'irritated flaky mess'), (4.0, 'wearing headscarves'), (4.0, 'skin flakes'), (4.0, 'showering process'), (4.0, 'shampoo bars'), (4.0, 'new ."'), (4.0, 'lot faster'), (4.0, 'limited time'), (4.0, 'couple weeks'), (4.0, 'bit strong')] | Irritated flaky, skin flakes, wearing headscarves |
| OMG where do i begin! worst purchase ever! scared me from ever buying a shampoo from lush ever again :( I use to have a normal scalp before using this. After using this 2-3 times a week, my scalp became sooo DRY!!! omg, after i finished this bar, i had the worst dandruff in my life :( i couldn't even live my normal day to day life without scratching the life out of my head every 2 minutes. This is not even an exaggeration. I lost a lot of hair in the process. Extremely disappointed and upsetting. I will say it does tingle when u apply it to ur scalp, but the cinnamon in the bar falls out after the 2nd or 3rd use. pointless. worst purchase i have ever made in my life. | [(23.0, 'scalp became sooo dry !!!'), (14.5, 'head every 2 minutes'), (13.0, 'day life without scratching'), (7.083333333333334, 'worst purchase ever'), (5.0, 'ur scalp'), (5.0, 'normal scalp'), (5.0, 'normal day'), (4.833333333333334, 'worst purchase'), (4.333333333333334, 'worst dandruff'), (4.25, 'lush ever'), (4.25, 'ever made'), (4.25, 'ever buying'), (4.0, 'u apply'), (4.0, 'extremely disappointed')] | Dry scalp, scratching, worst purchase |

Table 10. Full text of negative review, RAKE output, and negative themes

It was observed that for both positive and negative reviews, there are similar themes from different reviews. Positive reviews saw the themes of hair growth and noticing an improvement in hair quality while negative reviews saw themes of issues with sensitivity, and dry and itchy scalp. There is reason to believe that these themes influence customer product experience and can be considered when making business decisions regarding the products. From this small-scale thematic analysis, an easy-to-implement algorithm such as RAKE is still effective in uncovering themes.

**Conclusion and Future Work**

The process of building a machine learning model involves a significant amount of trialling and testing to determine which model is best for sentiment analysis. From trying different algorithms to testing different hyperparameters, it is clear that the optimal model is Logistic Regression with a C value of 10, an l2 penalty and liblinear solver. However, there are other methods of optimization that can still be applied. MNB, LR, and KNN algorithms were chosen as it was the deemed the most suitable from various literature review. Future work could include experimenting with other algorithms and observing its behaviour.

Data plays a crucial and probably the most significant role in any analysis. Data quality and quantity can significantly impact model performance as well. The LR model was built, trained, and tested using 774 rows of data which can be considered as a limited dataset. This was due to the nature of the data whereby there were more positive data than negative data to create a large yet balanced dataset. Model accuracy can be improved by adding more data samples. More information given to the model, the more it will learn and the more cases it will be able to identify correctly.

Once other optimization methods are applied to the model and more data samples are added, the model can be focused on deployment. This includes saving the model with pickle. Pickling is a process whereby a Python object hierarchy is converted into a byte stream, and unpickling is the inverse operation. Pickle module implements binary protocols for serializing and de-serializing a Python object structure[[11]](#footnote-11). Once the model is saved, anyone will be able to open the model using pickle and make predictions on unseen data.

Thematic analysis was conducted on a very small scale which means that there is room for improvement. As observed in the thematic analysis conducted, there were similar keywords found in different reviews which effectively uncovered both positive and negative themes. The process of thematic analysis could be further automated by determining these similar keywords and discarding the rest as there is a lot of unusable and redundant information from the current thematic analysis such as keywords with low scores.

By combining the sentiment analysis model with an automated thematic analysis process, this can streamline and optimize the process of reviewing and conducting analysis on customer product reviews. Sentiment analysis can also be applied to other areas such as customer service management. The application of automation and machine learning to business processes provides new and efficient ways of managing as well as improving a business. The outcome from sentiment and thematic analysis will hopefully allow Lush Singapore to solve its current business issues and serve as a guideline to create well-informed business decisions.

**References**

Brownlee, J. (2017, October 17*). How to Clean Text for Machine Learning with Python. Machine Learning Mastery*. https://machinelearningmastery.com/clean-text-machine-learning-python/

Isnain, A., Supriyanto, J., Kharisma, M. (2021*). Implementation of K-Nearest Neighbor (K-NN) Algorithm For Public Sentiment Analysis of Online Learning*. IJCCS (Indonesian Journal of Computing and Cybernetics Systems).

Medelyan, A. *Here’s why Thematic analysis beats sentiment analysis*. (2018, July 13). Retrieved August 17, 2022, from Thematic website: https://getthematic.com/insights/thematic-analysis-sentiment-analysis/#:~:text=Thematic%20analysis%20is%20more%20accurate,it%20lacks%20common%20sense%20knowledge

NLTK. (2009). Natural Language Toolkit — NLTK 3.4.4 documentation. Nltk.org. https://www.nltk.org/

Olagunju, T., Oyebode, O., & Orji, R. (2020). *Exploring Key Issues Affecting African Mobile eCommerce Applications Using Sentiment and Thematic Analysis*. IEEE Access, 1–1.

Oyebode, O., Alqahtani, F., & Orji, R. (2020a). *Using Machine Learning and Thematic Analysis Methods to Evaluate Mental Health Apps Based on User Reviews*. IEEE Access, 8, 111141–111158.

Oyebode, O., Ndulue, C., Adib, A., Mulchandani, D., Suruliraj, B., Orji, F. A., Chambers, C., Meier, S., & Orji, R. (2020b). *Health, Psychosocial, and Social issues emanating from COVID-19 pandemic based on Social Media Comments using Text Mining and Thematic Analysis* (Preprint). JMIR Medical Informatics.

Stecanella, B. (2019, May 10). *What is TF-IDF?* MonkeyLearn Blog. https://monkeylearn.com/blog/what-is-tf-idf/

Sharma, V. B. (n.d.). rake-nltk: Python implementation of the Rapid Automatic Keyword Extraction algorithm using NLTK. PyPI. <https://pypi.org/project/rake-nltk/>

Heidenreich, H. (2018, December 21). *Stemming? Lemmatization? What?* Towards Data Science. <https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8>

Band, A. (2020, May 23). *How to find the optimal value of K in KNN?* Towards Data Science. https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning : with applications in R*. Springer.

Mujtaba, H. (2020, September 24). *What is Cross Validation in Machine learning? Types of Cross Validation*. <https://www.mygreatlearning.com/blog/cross-validation/>

Brownlee, J. (2019, December 12). *Tune Hyperparameters for Classification Machine Learning Algorithms*. Machine Learning Mastery. https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/

**Appendix**

Sample of data source: [www.lushusa.com](http://www.lushusa.com) web pages

|  |
| --- |
| Graphical user interface  Description automatically generated |
| Graphical user interface, text  Description automatically generated |
| Graphical user interface, text  Description automatically generated |

Data collection: web scraper code

|  |
| --- |
| from gettext import find  from requests\_html import HTMLSession  from bs4 import BeautifulSoup  import requests  import pandas as pd  s = HTMLSession()  url = 'https://www.lushusa.com/bath/bath-bombs/rangoli-dreams/9999905921.html'  reviewlist = []  def get\_soup(url):      r = requests.get('http://localhost:8050/render.html', params={'url': url, 'wait': 3})      soup = BeautifulSoup(r.text, 'html.parser')      return soup  def getnextpage(soup):      pages = soup.find('div', {'class': 'pr-rd-pagination'})      if pages.find\_all('a', {'class': 'pr-rd-pagination-btn pr-rd-pagination-btn--next'}):          next\_page = soup.find\_all('a', {'class': 'pr-rd-pagination-btn pr-rd-pagination-btn--next'})          print(next\_page)          for element in next\_page:              url = element.get("href")              print(url)              return url      else:          return print("No URL found.")  def get\_reviews(soup):      reviews = soup.find\_all('div', {'class': 'pr-review'})      for item in reviews:          review = {      'title': item.find('span', {'class': 'pr-rd-review-headline pr-h2'}).text,      'rating': float(item.find('div', {'class': 'pr-rd-star-rating'}).text),      'body': item.find('p', {'class': 'pr-rd-description-text'}).text,          }          reviewlist.append(review)  while True:      try:          soup = get\_soup(url)          get\_reviews(soup)          print("Number of rows: ", len(reviewlist))          #get new URL          url = getnextpage(soup)      except:          print("No more URLs.")          break  df = pd.DataFrame(reviewlist)  df.to\_excel('rangoli-reviews.xlsx', index=False)  print("Fin.") |

Python code for sentiment analysis modelling and thematic analysis can be found in this link:

[Google Colab Link](https://colab.research.google.com/drive/1pso9qB4D6K9PoIYNhIp5Nz-8wfTVJmlh?usp=sharing)

1. From *A Beginner’s Guide to learn web scraping with python*, by Hiremath. O, 2022 (<https://www.edureka.co/blog/web-scraping-with-python/>) [↑](#footnote-ref-1)
2. From *Docker(software)*, by Wikipedia Contributors, 2022 (<https://en.wikipedia.org/wiki/Docker_(software)>) [↑](#footnote-ref-2)
3. From *scikit-learn*, by Wikipedia Contributors, 2019 (<https://en.wikipedia.org/wiki/Scikit-learn>) [↑](#footnote-ref-3)
4. From *Feature Extraction*, by Wikipedia Contributors, 2021 (<https://en.wikipedia.org/wiki/Feature_extraction>) [↑](#footnote-ref-4)
5. From *Supervised Machine Learning*, by Javatpoint Contributors, n.d. (<https://www.javatpoint.com/supervised-machine-learning>) [↑](#footnote-ref-5)
6. From *Naïve Bayes Classifier in Machine Learning*, by Javatpoint Contributors, n.d. (<https://www.javatpoint.com/machine-learning-naive-bayes-classifier#:~:text=Na%C3%AFve%20Bayes%20Classifier%20is%20one>) [↑](#footnote-ref-6)
7. From *Logistic Regression in Machine Learning*, by Javatpoint Contributors, n.d. (<https://www.javatpoint.com/logistic-regression-in-machine-learning>) [↑](#footnote-ref-7)
8. From *K-Nearest Neighbour(KNN) Algorithm for Machine Learning*, by Javatpoint Contributors, n.d. ( <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>) [↑](#footnote-ref-8)
9. From *Hyperparameter Optimization With Random Search and Grid Search*, by Brownlee, J. (2020, September 13). (<https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>) [↑](#footnote-ref-9)
10. From *Do I need to tune logistic regression hyperparameters?*, by Gusarov, M. (2022, April 19). (<https://medium.com/codex/do-i-need-to-tune-logistic-regression-hyperparameters-1cb2b81fca69>) [↑](#footnote-ref-10)
11. From pickle – *Python object serialization*, by Python.org Contributors, n.d. (<https://docs.python.org/3/library/pickle.html#:~:text=%E2%80%9CPickling%E2%80%9D%20is%20the%20process%20whereby,back%20into%20an%20object%20hierarchy>.) [↑](#footnote-ref-11)