City, University of London

MSc Data Science

Project Report

2022

Tracking a company's performance – a Natural Language Processing approach

Student:

Ethan Chew Wei Xun

Supervisor:

Dr. Vladimir Stankovic

21st December, 2022

Declaration

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed: **Ethan Chew Wei Xun**

Abstract

This study aimed to explore how Natural Language Processing ("NLP") techniques could be used to generate insights on the performances of a company's services in relation to its other offerings (internal monitoring) and against the market (benchmarking), in the context of the banking domain.

We designed and implemented an end-to-end NLP pipeline (consisting of unsupervised, semi-supervised, and supervised learning techniques) that produces a series of binary random forest classifiers and a multi-label transformer-based classifier from a clustered dataset comprised of unstructured customer reviews.

We iterated over a series of experiments, emphasising the relevance of clustering, and highlighting the role of expert judgement in potentially improving the quality of clustering. We also hypothesised the importance of mutual exclusiveness amongst clusters in generating better results.

We showed that by applying these models onto a text-based dataset with minimal processing, we could label text-based data with a taxonomy of topics without manual tagging and were able to get an indication of how different services within a company are performing. Through our approach, we illustrated N26's ongoing challenge with cryptocurrency-based services (over its other services) and Revolut's weak stock-based services relative to the market. Over an unseen dataset, we achieved a hamming score of 0.751.

Keywords: NLP, Multi-label Classification, Unstructured Data, Banking, Benchmarking

Table of Contents

1. Introduction and Objectives

As any industry matures, key market players are identified and cemented as giants within it. To capture the limited customer pool within the industry, tight competition arises with continuous innovation of new products and services (and improvements onto existing ones) often serving as key ingredients and centrepieces to not only customer acquisition but customer retention as well.

It is in a company's best interest to understand customer sentiment, needs, and implement proactive strategies (e.g. improving customer service, customer experience or user-interface of websites) rather than reactive ones. Supplementing it with outlooks on practices amongst competitors, companies can mimic (and develop upon) best practices and/or capitalise upon missed opportunities. Without either, even the biggest of giants will have to share the podium with emerging challengers or worse, fall.

Even in large industries, like UK's banking sector, similar narratives are seen for the former and the latter. The innovation and entrepreneurship drawn from customer insights paved ways for (initially) smaller companies to stand toe-to-toe with banking mainstays. A sector dominated by traditional banks like Barclays and HSBC, experienced tremors and disruption with the introduction of digital challenger banks like Wise and Starling Bank in 2011 and 2014 respectively – an introduction that was successful largely due to identifying gaps, and subsequently opportunities, in currency conversion and overseas transfers respectively. Contrastingly, laxed anti-money laundering policies led to failures and, subsequently, the reputational downfall of NatWest in late 2021 (Financial Conduct Authority, 2021); signalling a review exercise on anti-money laundering policies for other banks.

Forbes (2020) illustrated the "ultra-competitive market" banks face and alluded to the lack of customer loyalty, resulting in tight competition for customers. They highlighted challenges banks encounter in determining customer sentiments of its products and stressed, poor response rates of formal surveys (between 5% to 30%), and the importance of proactive strategies to minimise the churn rate of dissatisfied customers. Bain & Company (2018) concurred and emphasised the competitive markets, referencing the contention from tech companies, and accentuated the significance of quality as the driving factor in obtaining customer loyalty.

Customer satisfaction, sentiment, and needs provide insights for efficient prioritisation and inhibits innovation, allowing banks to solidify and potentially improve their market position. While inferences on customer needs (and satisfaction) can be drawn from internal private data (e.g. spending behaviour, time spent on webpages or frequency of visit), customer reviews on external platforms, like Trustpilot, may be a lower hanging fruit – despite the unstructured format of data.

We propose, developed, and implemented an end-to-end NLP pipeline to generate insights on a company's performance for internal monitoring, measuring, and benchmarking against its competitors and the market. We leveraged upon the immense volume of untagged informal customer reviews left

on Trustpilot, we developed a multi-label classification model (and pipeline) which took English banking reviews as inputs and outputted a series of labels corresponding to topics – customer service and account issues for instance. Though providers like Appen and Figure Eight offer annotation solutions through human intelligence (Appen, no date), we instead employed a cost-effective approach without manual tagging via a series of unsupervised learning techniques (e.g. word embedding, clustering and topic modelling). Transitioning over to semi-supervised learning, we produced a series of binary classifiers to generate a multi-labelled dataset; trained and evaluated through supervised learning techniques. Using a combination of a bottom-up and top-down approach, we built a banking taxonomy, from unstructured text, that is data-driven (bottom-up) and evaluated by expert judgement (top-down).

While this study worked within the banking domain, the chosen domain remains agnostic to the learnings and methodologies laid out within this study. Tailoring the dataset and expertise to the particular domain, companies are able to benefit from techniques within this study to garner insights on the performance of its services internally (against other services and offerings) and externally (benchmarking against competitors) – allowing for efficient prioritisation and gaining a marginal edge within the market. Players within the financial services industry with retail arms within the UK were used as a proof-of-concept but any company with an online footprint remain as beneficiaries to this study. Furthermore, unsupervised learning techniques laid out within this study provides an alternative to manual-tagging tasks (or aids it by conveniently supplying samples) for future NLP-based research tasks.

1.1. Objectives

This project aims to answer the following research question: **Can an NLP-based approach on entirely unlabelled data derive insights on potential trends and improvements to a company's offering and services?**

To address the research question, we established a set of objectives:

- Create and produce a dataset made up of unstructured text from English consumer banking reviews sourced from Trustpilot
- Explore and analyse the dataset to aid the data cleaning process, identify trends between (and within) banks and potential considerations as a result of structural differences
- Designing and employing a series of unsupervised learning techniques to identify a potential taxonomy of topics within the domain
- Utilise semi-supervised learning techniques to develop and extend a series of binary classifiers to generate a multi-labelled dataset
- Apply supervised learning techniques to train a multi-label classifier
- Experiment over different hyperparameters, inputs, and evaluating differences in performance

- Prove and obtain insights of company performances through the implementation of work products

The success and failures of objectives are evaluated through a combination of explicit metrics (e.g. accuracy measures) and/or expert judgement (contingent on the appropriateness and availability) assessed and presented in an unbiased manner – highlighting the successes and caveated by potential considerations.

The final work products of this study are a collection of binary classifiers and single multi-label classifier which takes the review (sentence embeddings) as input and outputs a sequence of labels, corresponding to a taxonomy of topics surrounding the banking domain. Examples of insights are presented and discussed, along with the performance of models – both, binary and multi-label classifiers.

1.2. Work Plan

The project plan of this study follows closely to the high-level work plan presented in Figure 1. A detailed breakdown of the end-to-end pipeline is presented in Section 3.

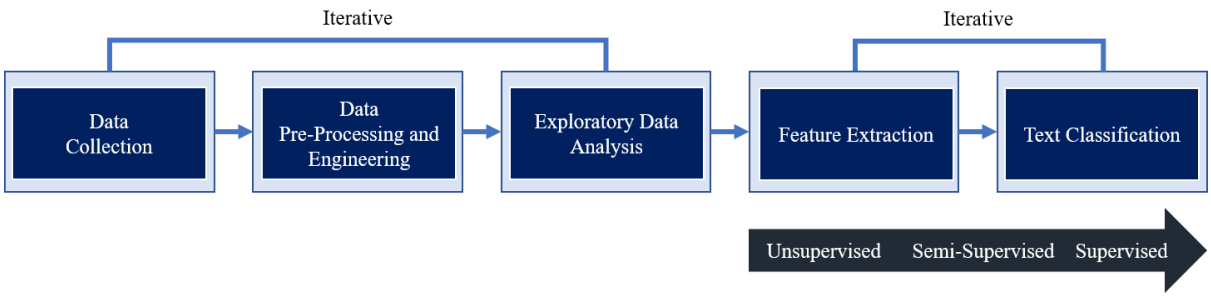


Figure 1: Overview of end-to-end pipeline

Aligning to the pipeline components, we adhered to four key milestones:

- Milestone 1 – iterative completion of data collection, processing, exploration and analysis with continuous evaluation and improvements
- Milestone 2 – iterative completion of feature extraction and text classification, with continuous evaluation, experimenting (with different inputs and hyperparameters), and intermediary results
- Milestone 3 – on-going referencing and literature review throughout the project, up to the implementation of each iterative block of components
- Milestone 4 – drafting and finalisation of report

The project plan was generally conformed to, with minimal changes in the designed pipeline. Components like feature extraction and multi-label classification development demanded more time than expected but flexible and conservative planning allowed for rearrangement of deliverables.

1.3. Report Structure

This section introduced the importance of customer sentiment in customer acquisition and retention. We outlined a proposed approach to leverage upon the rich number of reviews left on consumer review website like Trustpilot and discussed the potential insights from the work products of this study – a way for companies to evaluate the performance of its products and benchmark against its competitors. We highlighted the beneficiaries to this study, objectives to this study and work plan to maintain accountability.

Section 2 provides context to the work that was carried out, covering the technical material and methodology this study lent from. We cover past works that has been performed with multi-labelling and transformer-based models in particular. Inspirations for each component in the pipeline are discussed, focussing on the technicalities and how each approach addressed the challenges found in preceding approaches.

Section 3 explains our designed pipeline in detail, articulating each component – including the goals, implementation details, library dependencies, and iterative process, if any. Starting with data collection and ending at multi-label classification, the sections dictates the input and output to each component and the dependencies of the subsequent component. The approaches within each component are defined and summarised through an illustrated example at the end of each sub-section, if necessary.

Section 4 mirrors the details of Section 3 but presents the implementation results to each component (and approaches within it) – including any experimental and intermediary results. Implementation details are reinstated and critical evaluation to the results are provided. The results of each section vary from distributional plots to explicit performance metrics, depending on the case and appropriateness.

Section 5 evaluates the results from Section 4 in the context of the objectives listed in Section 1. We reflect upon the learnings from the literature review, answer our proposed research question and assess the success and failures of our objectives. We address the performance of our work product (e.g. user satisfaction) and key elements that may bolster it.

Section 6 reflects upon the work carried out as part of this study – choice of objectives, literature examined, methods used, and efficacy of the work plan. We summarise the key findings, learnings, and present the contributions of the project. We conclude our study by discussing potential opportunities, that appear to be good candidates for future work, and effectiveness of implementation details (in hindsight).

2. Context

This section presents the existing state of knowledge based on the literature review. We discuss the current state of multi-label text classification and motivations for our proposed approach. Subsequently, we present various techniques applicable to our study, sub-sectioned by the corresponding component.

Kowsari *et al.* (2019) summarised and presented the current state of text classification algorithms, highlighting common pre-processing techniques, the advantages and limitations of text classification techniques. They discussed the ease of implementation of techniques like logistic regression but caveats its performance by its linear property (inability to solve non-linear problems). Conversely, they showcased the flexibility and performance of deep learning approaches but at the expense of a black-box model and computational intensity.

Regardless, deep learning approaches like convolutional neural networks (CNN) have found applications in text classification, achieving novel results against linear and other deep learning (e.g. recurring neural networks (RNN)) approaches (Wang, Huang and Deng, 2018). Even a simple neural network (with added complexities like rectified linear units (ReLU) and dropout) found success in text classification over an array of datasets (Nam *et al.*, 2014). Zhou et al. (2015) combined the strengths of CNNs and RNNs to produce a unified model C-LSTM – an approach built upon CNNs and a long short-term memory recurrent neural network (LSTM). Over question classification tasks, they found C-LSTM to outperform CNN, LSTM, and accuracies of 94.6% overall.

However, since the introduction of the transformer architecture (Vaswani *et al.*, 2017) and subsequently Bidirectional Encoder Representations from Transformers (BERT) (Devlin *et al.*, 2019), BERT has remained as the state-of-the-art approach and the default for several NLP tasks including text classification (Gonzalez-Carvajal and Garrido-Merchan, 2021). Its implications spread over a wide range of domains, including innovations in the biomedical field for chemical identification and multi-label classification over PubMed articles – birthing new state-of-the-art models specific to the biomedical domain was achieved.

While studies on the application of text classification algorithms supplied stellar results, the studies utilised a labelled dataset or a subset of it. To our best knowledge, we did not find a study which made use of an entirely unlabelled dataset for (multi-label) text classification. Thus, we propose an approach which employs a combination of unsupervised, semi-supervised, and supervised learning techniques. Unsupervised learning techniques are used to cluster and assign topics to clusters in an autonomous manner without manually assigned labels. Semi-supervised learning techniques builds binary classification models over a subset of data points. Supervised learning takes a now fully labelled dataset and fits a multi-label classification model.

2.1. Data Pre-Processing and Engineering

In their works of text classification over 'Twitter' tweets, Sriram *et al.* (2010) discussed common and traditional data pre-processing techniques applied across domains to reduce the dimensionality of feature vectors. They explained the process of tokenisation as a way to dissolve a document into its words (or sentences) and the method of stemming which reduces words to its root form – the example of reducing the words: 'running', 'ran', 'runner', and 'runs', to 'run', was given. They covered additional generic approaches like stop word removal, whereby words that do not provide significant value to the semantic of a sentence or document is removed, before advising on the need for a deep understanding of the domain. Working with tweets, they identified tweet-specific pre-processing steps to take (e.g. removal of frequent '@' character) and trends within classes of tweets.

Howard and Gugger (2020) employed a different approach to data pre-processing during their implementation of a deep learning library, 'fastai'. They expressed the importance of each element within a sentence and raised a trade-off between simplifying the representation of text and the loss of information as a result of it – citing the potential loss of sentiment and semantic from simplifying the number of tokens through the process of lower-casing text. Solving both, they utilised special tokens in replacement of and to represent common data pre-processing techniques. For example, 'xxmaj' depicts upper-cased letters and are placed in front of the (originally) capitalised letter – "Hello" to "xxmaj hello". Other special tokens included ones for padding, all-caps words, and repeated words/characters.

Edunov *et al.* (2018) discussed the challenge with sparse real-world data – especially within a specific and niche domain. They proposed and explained the idea of back-translation whereby synthetic data points are generated through translating a sample of text to a different language and back to the original language; selecting the best output (largest estimated probability) using searching algorithms, commonly beam or greedy search. Supplementing the training with the synthetic translations, they found optimal results when performing back-translation on sampled or noised-applied signals. Using BiLingual Evaluation Understudy (BLEU) as a metric to evaluate the quality of translated text, they achieved state-of-the-art results over an English-German dataset.

2.2. Word Embeddings

Bag-of-Words (BOW) approaches provide a convenient way of understanding the importance of words relative to a corpus. Words within the corpus form a dictionary and occurrence frequencies of each word within a document are recorded within a matrix, forming a document term matrix. Term Frequency (TF) refers to the frequencies of each word (or term) while Term Frequency – Inverse Document Frequency (TF-IDF) represents the importance of words, within a document, through the scaling of word frequencies by its corresponding frequencies within the entire corpus (Rajaraman and Ullman, 2011). Though TF-IDF proves to be a popular approach with 83% of use throughout text-based recommender systems in digital libraries (*Beel et al.*, 2016), traditional BOW approaches often run into

challenges with sparsity (due to presence of infrequent words) and the preservation of semantics due to the disregarding of word orderings (Mikolov *et al.*, 2013).

Contrastingly, word embedding techniques aim to capture semantic meaning of words under the assumption that words which share the same context are semantically similar (Mandelbaum and Shalev, 2016). Mikolov *et al.* (2013) proposed two new word embedding model architectures for word representation, commonly referenced as "word2vec". They reiterate the benefits of neural networks in the preservation of linear regularities among words and introduce a Continuous Bag-of-Words (CBOW) and Continuous Skip-gram (Skip-gram) model as alternatives and improvements to vector representations of words. Both utilise a feedforward neural net language model (NNLM) but CBOW predicts the word based on its context (i.e. surrounding words) while Skip-gram does the opposite and predicts the surrounding words based on a single word. From it, they found improvements in accuracy, computational intensity, and contemporary performances over similarity tasks - syntactic and semantic.

As an improvement to both, Pennington, Socher and Manning (2014) instead utilised Global Vectors for word representation, or commonly referred to as "GloVe". They highlight the poor word analogy tasks produced by LSA and narrow view of word2vec models (working with only a series of small local context of words). Instead, GloVe works on a word-word co-occurrence matrix over the entire corpus and displays it in a probabilistic manner. A word-word co-occurrence matrix maps the frequency of pairs of adjacent words (word-word) in a corpus, represented by a square matrix. They found GloVe to outperform its counterparts, like SVD and CBOW, for word analogy and similarity tasks over different hyperparameters.

Cer *et al.* (2018) stresses the lack of data that exists within NLP tasks and emphasises the frequent reliance on transfer-learning and pre-trained word embedding models (like word2vec and GloVe) to work around this limitation. Instead, they present two novel models for generating sentence embeddings that demonstrates generalisable behaviour to a suite of NLP tasks and datasets, even over a small dataset. Marketed as universal sentence encoders, they introduce two encoding models - deep averaging network (DAN) and transformer-based approaches.

The DAN-based uses an averaged set of word embeddings (unigram and bigrams) and deep neural networks (DNN) to produce 512-dimensional sentence embeddings. Conversely, the transformer-based approach leverages upon the sub-graph of the transformer architecture to produce context aware word representations. Through attention, at any word within the sentence, a view on the ordering and identity of all other words can be obtained (context aware); allowing for parallelisation and thus, reduced training times (Vaswani *et al.*, 2017). At each word position, the element-wise sum of the representations is derived; forming a sentence encoding vector of fixed length which is subsequently scaled down by the length of the sentence.

They found the transformer-based approach to generally outperform the DAN-based approach, though at the cost of increased resource intensity. Attributing it to the difference in time and space complexity, they explain how the time and space complexity of the transformer-based approach are $O(n^2)$ (while $O(n)$ and $O(1)$ for the DAN-based approach respectively). Additionally, they discovered transformer-based universal sentence encoders to perform as well as other approaches (word2vec) using pre-trained embeddings and larger sample sizes - sample sizes of 1,000 and 67,300 respectively.

2.3. Dimensionality Reduction

High-dimensional data poses challenges with visualisation, computation, and are often reduced prior to preceding processing (Sorzano, Vargas, and Pascual-Montano, 2014). Dimensionality reduction aims to produce low-dimensional representations of, originally, high-dimensional data – bringing the data from a high-dimensional to a low-dimensional space. Sorzano, Vargas, and Pascual-Montano (2014) discussed two overarching approaches to low-dimensional representations: one which keeps the most relevant variables (feature extraction) and another which excludes redundancies and presents equivalent amount of information with a smaller set of variables through the combination of input variables (dimensionality reduction).

Jolliffe (2002), introduced a method for dimensionality reduction – Principal Component Analysis (PCA). PCA aimed to identify a set of deterministic number of axes (principal components) which captured the most variation within the data. By taking linear combinations of variables, the original features are transformed into a smaller set of orthogonal and uncorrelated variables – principal components. Principal components are ranked in order of the variation it captures and is determined by minimising the perpendicular distances from input values to principal components.

However, van der Maaten and Hinton (2008) address shortcomings of PCA. They argued that PCA focusses on separating low-dimensional representations of dissimilar points rather than inching similar points closer together – a larger priority for high-dimensional data that sits on (or close to) low-dimensional and non-linear manifolds. They attributed the unattainable outcome to the linear properties of PCA (i.e. linear mapping) and highlighted the challenges of existing non-linear approaches when working with real and high-dimensional data. Specifically, they described a trade-off between local and global structure representations and the absence of an approach which attained both under a single mapping. Instead, they introduced t-distributed Stochastic Neighbour Embedding (t-SNE) - a non-linear approach to dimensionality reduction.

t-SNE is built upon its predecessor, Stochastic Neighbour Embedding (SNE), but accounts for the crowding problem commonly found in SNE. Both approaches depict similarity between points through a conditional probability distribution (Gaussian and t-distribution for SNE and t-SNE respectively), representing the probability that a point, j, would pick another point, i, as its neighbour. Pair-wise points which are close together are shown through high conditional probabilities while conditional probabilities

of those which are further apart are smaller. They argued that due to the short tails of a Gaussian distribution, it creates a crowding problem where distances over neighbourhoods of points are not preserved. To circumvent this, they instead used a t-distribution with a heavy-tailed distribution instead. Testing it over a wide array of data, they found it to outperform other non-parametric approaches like Isomap and Sammon mapping.

McInnes, Healy, and Melville (2020) however, raised the drawback of t-SNE. They touch on challenges with scalability (onto larger datasets), computational intensity, and lack of global structure preservation with t-SNE. As an improvement, they introduced Uniform Manifold Approximation and Projection (UMAP) built upon nearest neighbour graphs, grouping it under the same umbrella as t-SNE. They highlighted the improved preservation of global structure and computational intensities - citing the improved run-times and generalisability from the lack of computational restrictions on embedding dimensions. Similar to t-SNE, UMAP is simplified into two phases: construction of a weighted neighbour graph and projection onto a low-dimensional space.

With the former, an approximation to the topology of the data is obtained by connecting data points to its neighbouring points. By extending a radius and forming a neighbourhood around each point, nearest neighbours are identified and connected. However, using a fixed radius results in shortcomings when stumbling upon low density regions with no nearest neighbours. Citing an example from their github (lmcinnes/umap, no date), we see this problem illustrated as shown in Figure 2.

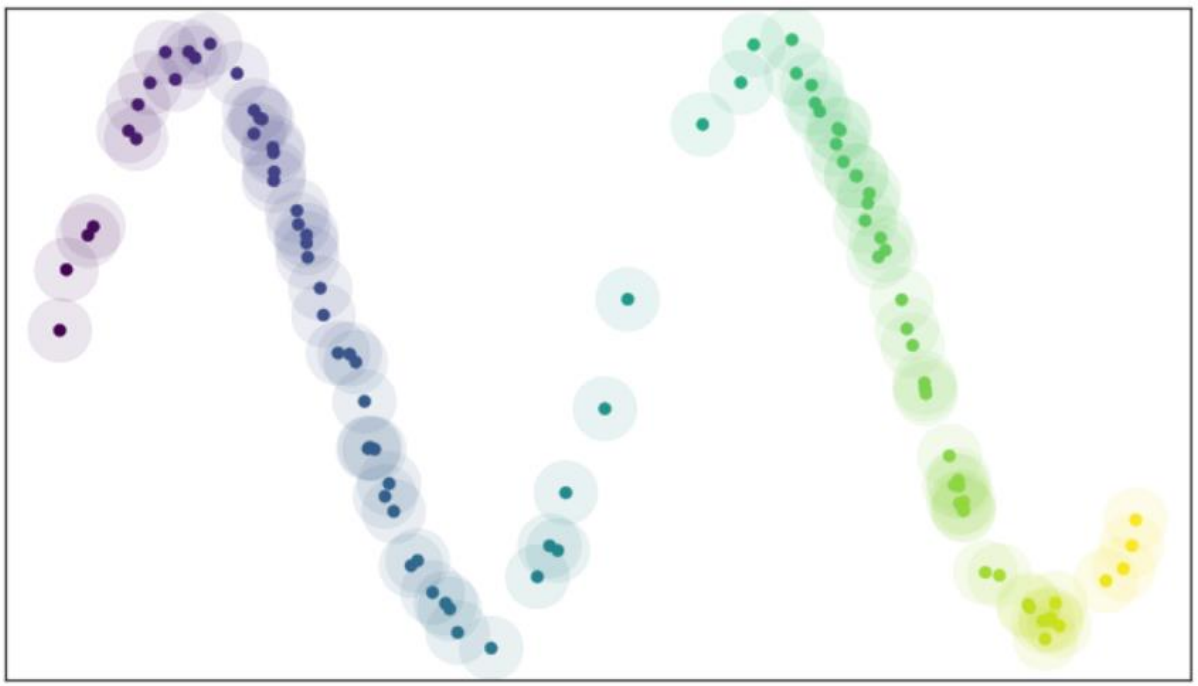


Figure 2: UMAP example - challenges with a fixed radius (lmcinnes/umap, no date)

With the given radius size, teal points fall within low density regions with no nearest neighbours. Instead densities surrounding points are proxied by a hyperparameter, k (or n_neighbours), and a varied

radius is casted from each point, with the size of the radius being inversely proportional to its density. In essence, larger radii are casted around points in low density regions while smaller radii are extended around points in higher density regions; leading to representation as shown in Figure 3.

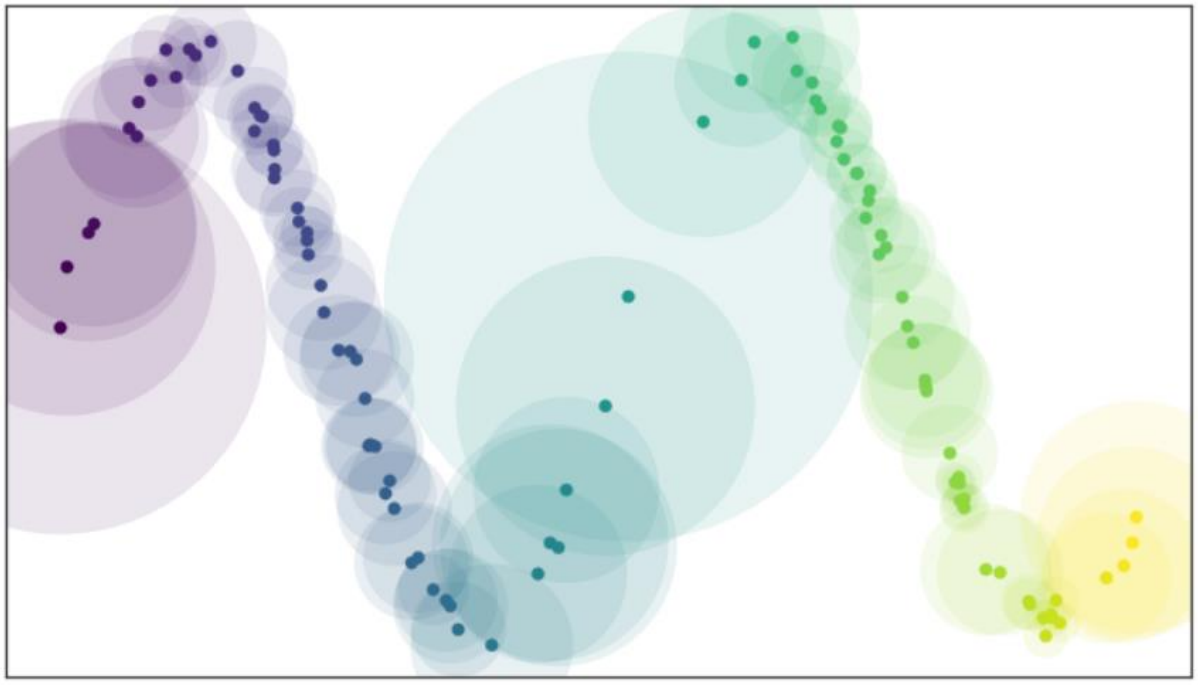


Figure 3: UMAP example - varying radii sizes (lmcinnes/umap, no date)

Varying radii sizes enable points in low density regions to identify a set of neighbours. High density regions are shown through smaller radii and significant amounts of overlap. From it, points and its k-nearest neighbours are connected, forming a k-neighbour graph. The edges between points are assigned weights as a representation of its respective connection probabilities – points which are closer together are assigned higher connection probabilities while the converse is true for points which are further apart. Figure 4 illustrates these connections through a weighted k-neighbour graph and weights of the edges are symbolised by the saturation of the colours – high levels of saturation correspond to higher connection probabilities. The trade-off between global and local structure is determined by the size of k.

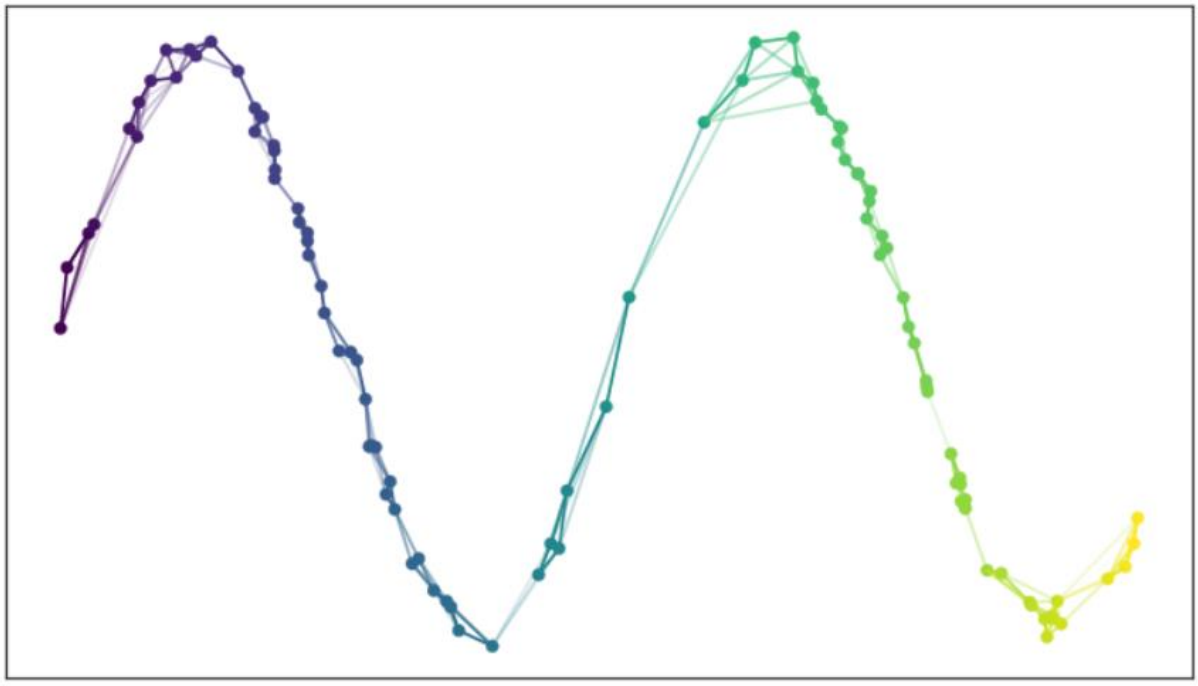


Figure 4: UMAP example - weighted k-neighbour graph (lmcinnes/umap, no date)

Projecting the weighted k-neighbour graph onto a low-dimensional representation, the weight of edges determines the spatial relationships between points. Pair-wise points with highly weighted edges are more likely to be grouped together while lowly weighted edges are more likely to be separated apart.

2.4. Clustering

With real-world classification tasks, the large presence of unlabelled data often serves as a challenge to obtain granular insights - unsupervised learning. However, clustering techniques serves as a way to identify underlying patterns and/or structure within datasets (Alashwal *et al.* 2019). Macqueen (1967) introduced a parametric approach to clustering by grouping data points into k number of clusters – k-means clustering. As a parametric approach, k acts as a hyperparameter that is specified and represents the prior-beliefs of the data. At inception, k number of centroids are randomly assigned and are treated as clusters. The remaining points are allocated to the closest cluster (determined by the Euclidean distance) and the centroids (mean) of each cluster are recomputed. By iteratively allocating points to clusters, recomputing the centroids, and terminating the process when there are no updates to the centroids, k number of clusters are formed. Macqueen, however, caveats this process by the initial assignment of centroids. They indicated that the quality of clustering is dependent on the initial assignment of centroids and suggested repeated experiments to mitigate this risk.

Campello *et al.* (2013) generalised clustering algorithms to fall under an orthogonal set of features: hierarchical and flat, and centroid and density-based - common approaches like k-means clustering is flat and centroid-based while other novel approaches like Density-based spatial clustering of applications with noise (DBSCAN) are flat but density-based. They discussed on a set of limitations

that exist within other approaches. They mentioned how flat clustering algorithms may lead to erroneous clustering, given the single density threshold, and how existing hierarchical approaches fail to allow for easy interpretation of most significant clusters. Instead, they introduced a new hierarchical and density-based approach for clustering: Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). HDBSCAN forms clusters in accordance to the density (or sparsity) of points and uses a bottom-up approach to group clusters (forming a hierarchy) in accordance of their distances. Improving upon it, McInnes and Healy (2017) presented an accelerated algorithm approach to HDBSCAN and developed an open-source library for the implementation of it.

They simplified the process of HDBSCAN into a series of steps. Campello *et al.* (2013) represented the distance of a point, x, to its kth nearest neighbour as its core distance – denoted as $core_k(x)$. Defining a distance function between a pair of points, a and b, their mutual reachability distance compares the core distances of a, b, and the distance between them; as denoted in equation [1].

$$d_{mreach-k}(a,b) = max\{core_k(a), core_k(b), d(a,b)\} \quad [1]$$

The mutually reachability distance takes the maximum of the three distances and are illustrated as an example, recited from McInnes (scikit-learn-contrib/hdbscan, no date), in Figure 5.

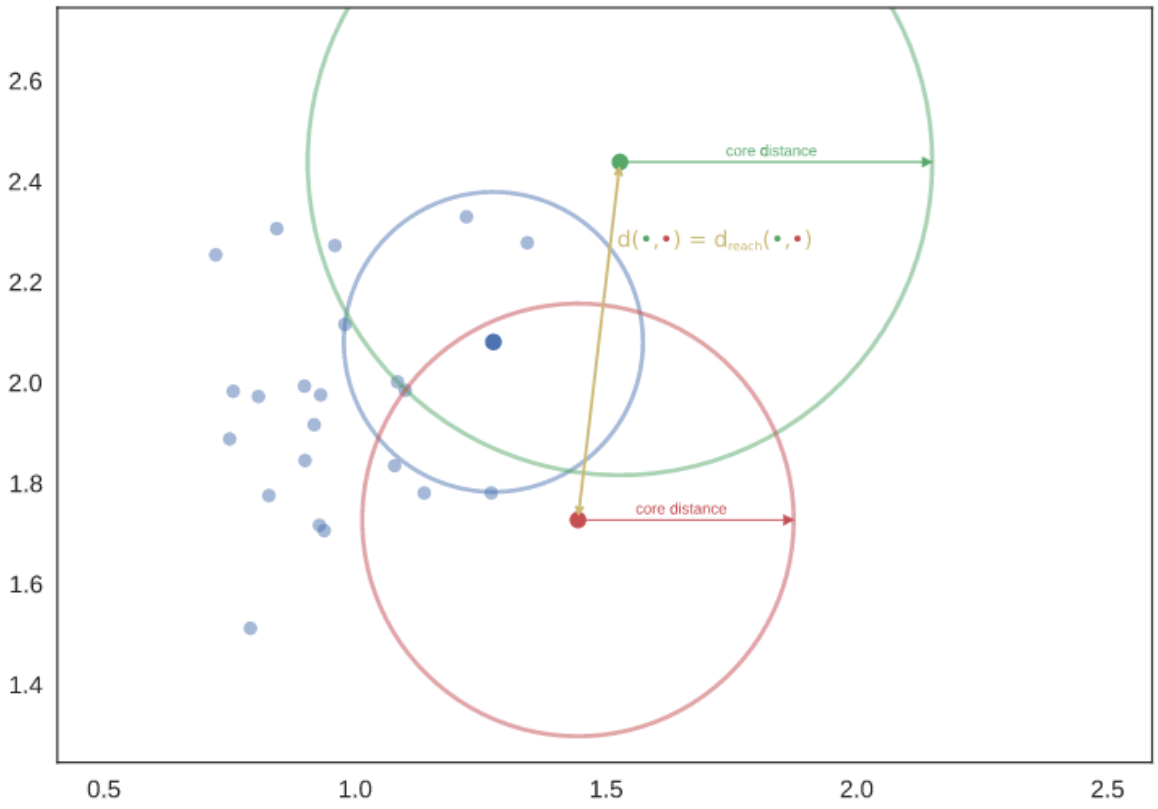


Figure 5: HDBSCAN example - mutual reachability distance (scikit-learn-contrib/hdbscan, no date)

The illustrated example takes k to be 6, resulting in a neighbourhood which encapsulates 5 other points. The mutual reachability distance between the red and green points is the distance between both points (i.e. $d(a, b)$) since it exceeds the neighbourhood of both (i.e. being larger than the core distances of both). Conversely, the core distance of the red point is taken as the mutual reachability distance between it and the blue point.

Viewing data points as vertices and mutual reachability distances as weighted edges, with smaller distances being equivalent to lower weights, a minimum spanning tree is built using a greedy algorithm. In essence, a single edge is added at a time, taking the lowest weighted edge that connects an unconnected vertex. By doing so, a hierarchical representation can be produced by steadily lowering threshold values and dropping any edges that falls above it – essentially disconnecting the minimum spanning tree and forming disjointed clusters. The generated minimum spanning graph over the same example is given in Figure 6.

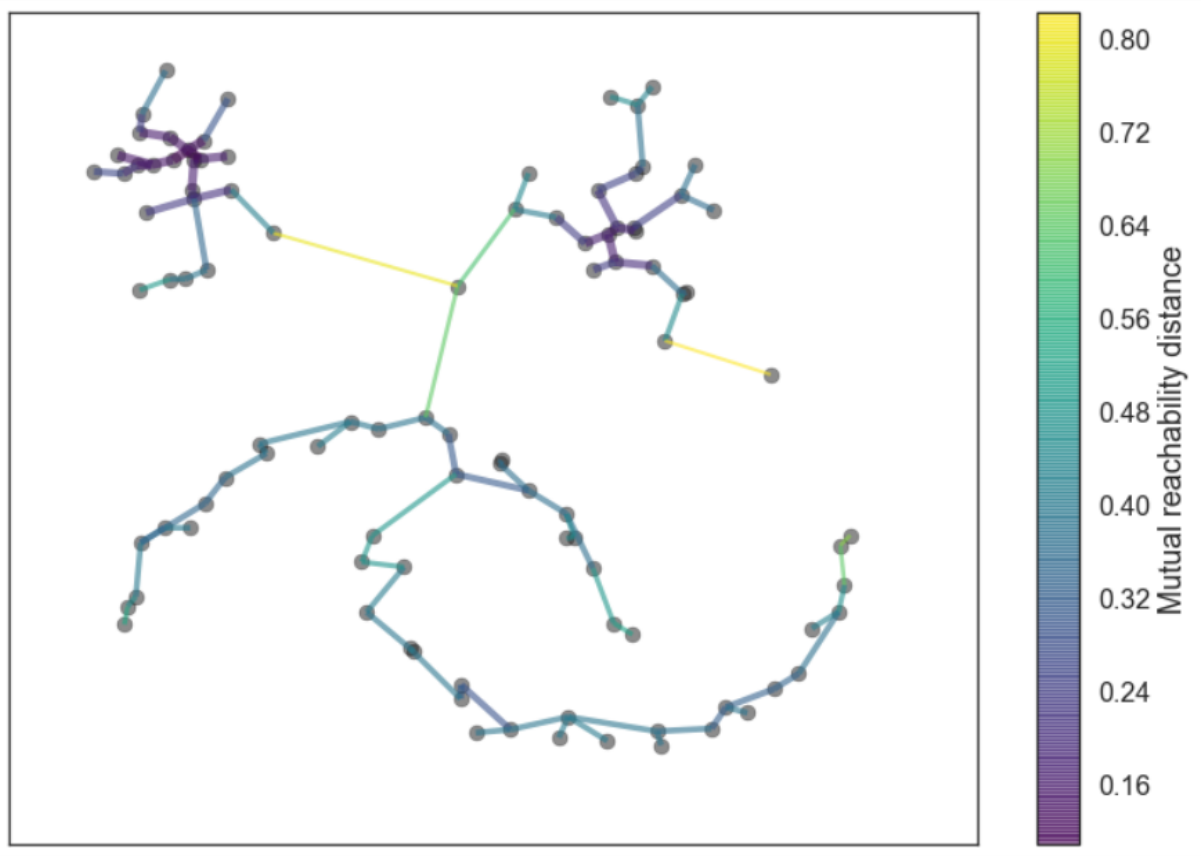


Figure 6: HDBSCAN example - minimum spanning graph (scikit-learn-contrib/hdbscan, no date)

Employing a bottom-up approach and sorted order of edges, merged clusters are created for each edge. Through the use of the 'find' function that exists within union-find data structures, neighbouring clusters are identified and merged. Plotting the results onto a dendrogram, granular views on the hierarchical structure can be obtained, as shown in Figure 7.

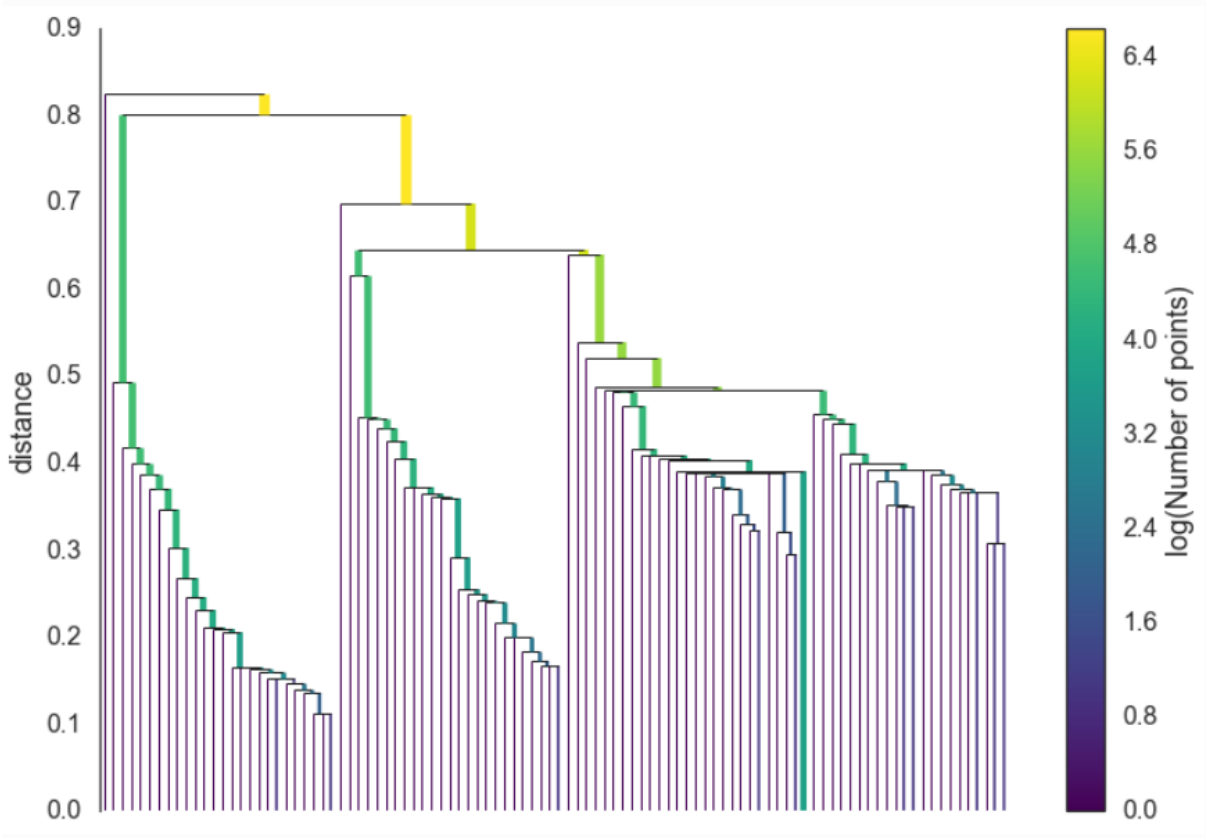


Figure 7: HDBSCAN example – granular dendrogram (scikit-learn-contrib/hdbscan, no date)

The granularity of the dendrogram, however, serves as a double-edged-sword – providing detailed views on the hierarchy but adds complexity through a complicated hierarchy. Additionally, preferences on cluster granularity introduces redundancies onto granular dendrograms – a higher-level view may be preferred. Instead, the cluster tree is condensed by declaring a minimum cluster size (min_cluster_size) which controls the number of points required to constitute towards a cluster – ultimately dropping clusters which have fewer points than the minimum threshold. A variable, λ, is introduced to keep track of the distances where clusters fall-out and is taken to be the inverse of the distance. Using a top-down approach and steadily increasing the λ (decreasing the distance), parent clusters are split into smaller clusters and a condensed (and simpler) dendrogram is produced as shown in Figure 8.

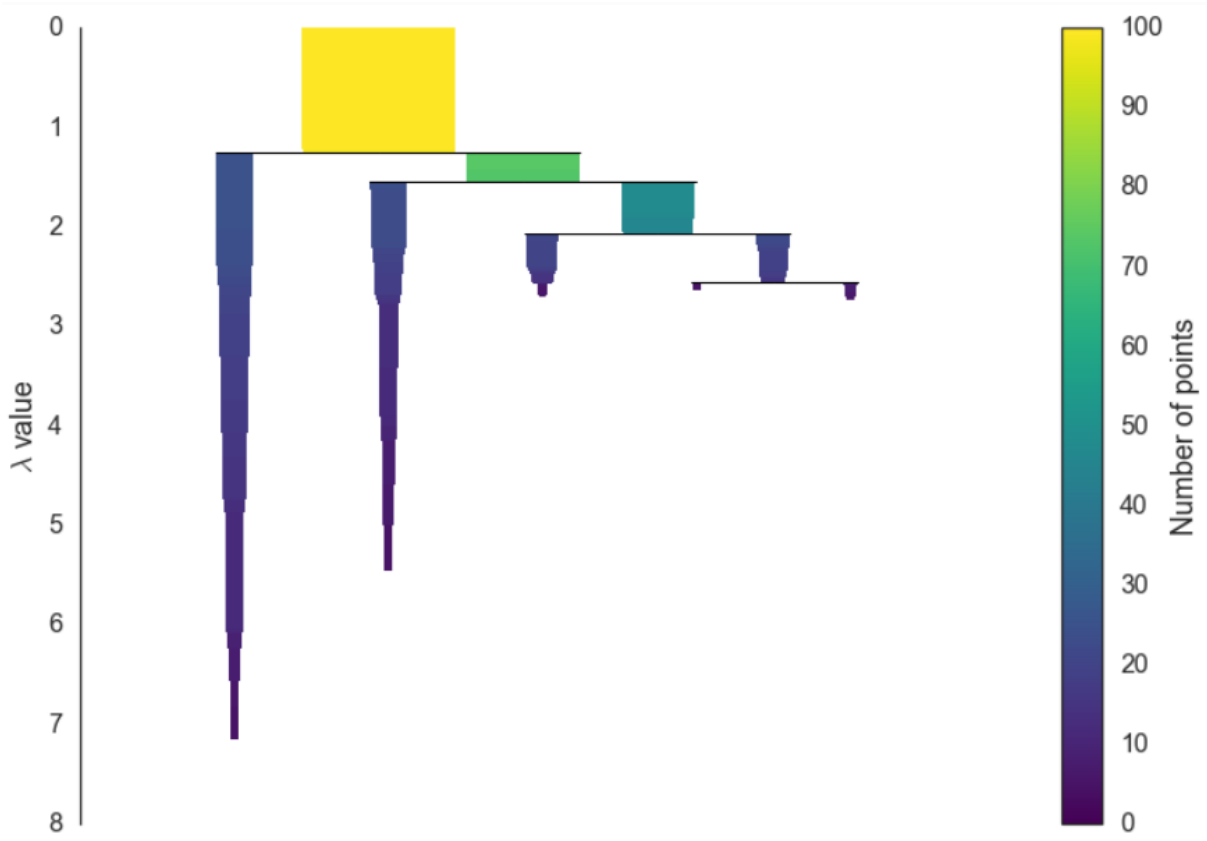


Figure 8: HDBSCAN example - condensed dendrogram (scikit-learn-contrib/hdbscan, no date)

While Density-based Spatial Clustering of Applications with Noise (DBSCAN) uses a fixed distance measure to extract the clusters (represented by a horizontal line cutting the dendrogram and selecting the clusters which intersect it), HDBSCAN instead extracts the clusters with the highest stability. Stability is measured by computing the λ where the cluster originates and when it falls off. Visually, these are clusters which persist the longest (by distance) – essentially, the clusters with the largest area on the dendrogram. As seen in Figure 9 , the circled clusters are extracted.

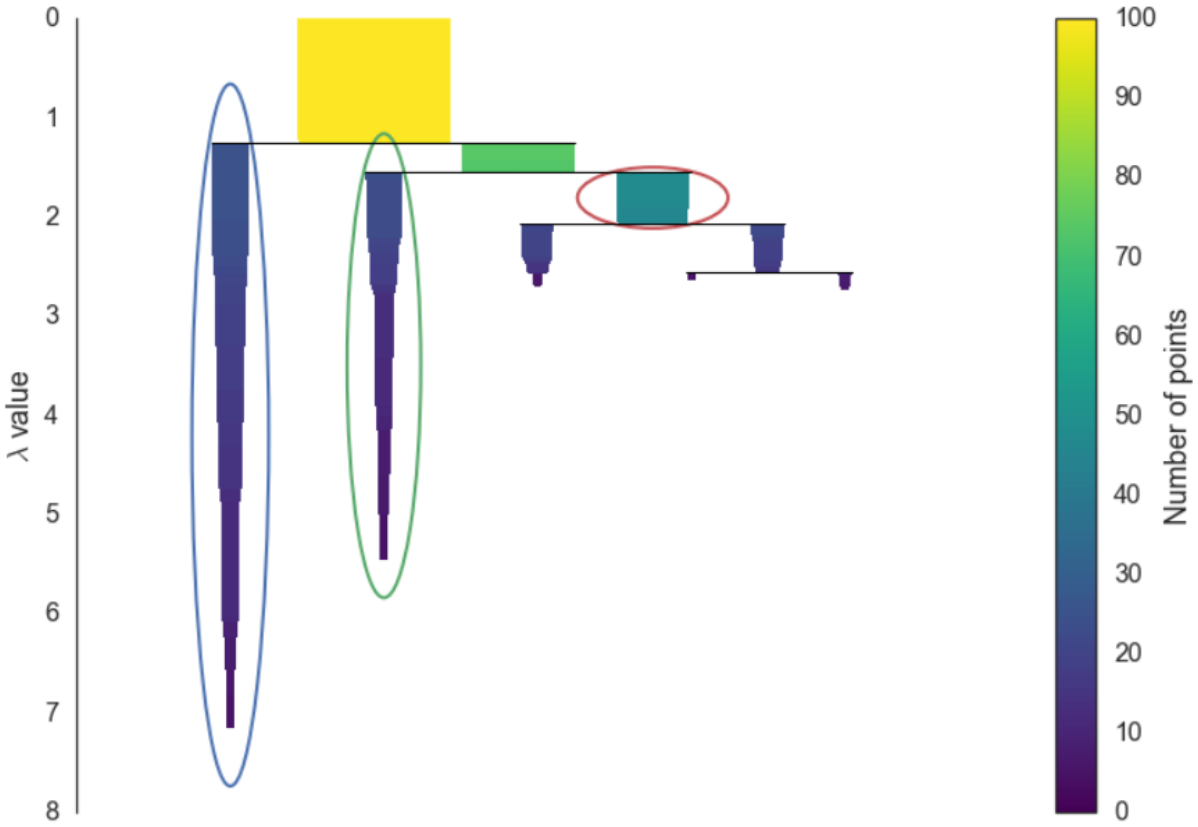


Figure 9: HDBSCAN example - extracted clusters (scikit-learn-contrib/hdbscan, no date)

Applying it over a range of datasets and domains, Campello et *al.* (2013) found HDBSCAN to outperform its density-based clustering counterparts for a majority of them. Reiterating the performance of HDBSCAN, McInnes and Healy (2017) found it to produce comparable results to DBSCAN but with the added convenience of hyperparameter tuning and support over variable density clusters.

2.5. Topic Modelling

To understand the semantics of text-based data, topic modelling approaches have been developed and provide a way to identify a topic (or set of topics) over a set of documents. Landauer, Foltz and Laham (1998) introduced such an approach – Latent Semantic Analysis (LSA). They explained how LSA as an approach does not solely rely on co-occurrence matrices, not manually constructed taxonomies or dictionaries, but instead applies singular value decomposition (SVD). From a word-document co-occurrence matrix (M), SVD is applied to decompose into a series of three components: a word-topic matrix (T), topic-document matrix (D), and diagonal matrix of scalars (Σ) representing the topic strengths. M is deconstructed such that $M = T\Sigma D^T$. Selecting s number of topics, the first s number of dimensions are considered for Σ (D and rows of T) word similarities and topics are derived through the reconstruction of M_s.

As an alternative approach, Blei, Ng, and Jordan (2003) introduced a probabilistic approach to topic modelling – Latent Dirichlet Allocation (LDA). In summary, LDA outputs a pre-determined

number of topics by evaluating two sets of probability distributions: the distribution of topics over documents and distribution of words over topics. It is built over the underlying assumption that documents with similar topics will use a similar group of words. They conjected that documents are probability distributions over latent topics (i.e. uncovered topics) and topics are probability distributions over words. To put it simply, documents are made up of a number of topics and topics are made up of words which are commonly used together. They summarised the process of LDA and dependencies of model parameters into an illustrated cited in Figure 10.

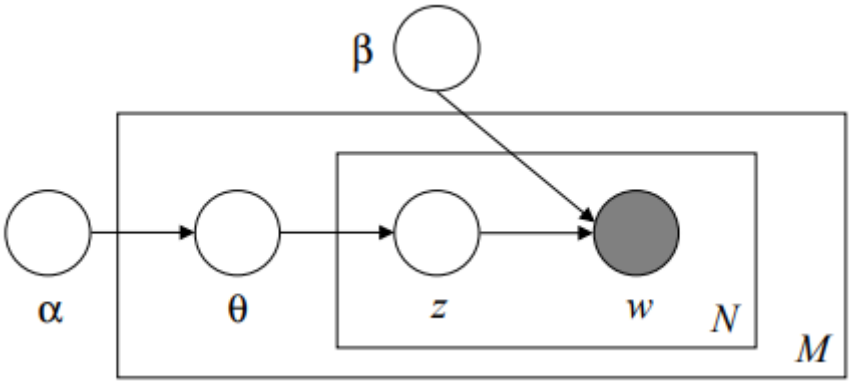


Figure 10: Illustrated representation of LDA (Blei, Ng, and Jordan, 2003)

Encasing rectangles show model dependencies over number of documents, M, and words, N. α and β represents the Dirichlet-priors to the per document topic distribution and per topic word distribution respectively. Higher values of both signify a larger mixture of topics per document and words per topic respectively. θ indicates the topic distribution for each document, m, and z notates each topic for each word, w. Mathematically, the dependencies of each parameters are shown in equation [2].

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \quad [2]$$

In essence, the joint distribution, given parameters α and β, of a topic mixture θ, set of N topics z, and set of N words is given by the product of the conditional probabilities for topic z (over a given mixture θ) and word w (over a given topic and β) across all words (and multiplying on the conditional probability of mixture θ given α). After repeated computations on the joint distribution across each document, a steady state of topic assignments is reached. Applying over document modelling, classification, and collaborative filtering tasks, they found LDA to outperform other approaches like latent semantic indexing (LSI) and probabilistic latent semantic indexing (pLSI).

Lee and Seung (2000) introduced generic algorithmic approaches for Non-negative Matrix Factorisation (NMF) and later repurposed by Kuang, Choo, and Park (2014) in the context of topic modelling and document clustering. NMF provides an approach to decompose a non-negative matrix V, to the product of two non-negative matrix factors, W (a lower dimensional representation of V) and H. Essentially, V is approximated by W and H ($V \approx WH$) and is found through minimisation of the

approximation error $||V - WH||^2$. Through iteration over two multiplicative update rules (alternating between H and W), the approximation of V converges. In the context of topic modelling, similar to LDA, W forms basis vectors and produces word-topic representations while H forms a coefficient matrix comprised of membership weights for documents relative to each topic. The basis vectors serve as topics and the number of topics is exogenous and pre-defined, as seen in LDA.

2.6. Evaluation

Evaluation metrics provide tools to validate and select best performing methods (e.g. clustering). Dudoit and Fridlyand (2002) categorised evaluation methods into two broad categories: external and internal indices or measures. With the former, they explained how external measures often required a set of ground truth labels or known cluster labels for evaluation – an approach suited for supervised or semi-supervised learning techniques whereby labels are accessible over the entire or a subset of the dataset. Caruana and Niculesu-Mizil (2006) summarised supervised learning algorithms and compiled a suite of objective external performance metrics. Highlighting metrics like accuracy, f-score, and the area under the Receiving Operating Characteristic curve (ROC), they emphasised the importance of considering a varied range of performance metrics to amass different views on the successes and failures.

Contrastingly, Dudoit and Fridyland (2002) discussed internal measures which are frequently subjective and are derived from the same observations used for the given unsupervised learning approach like clustering. Moulavi *et al.* (2014) reiterated the challenges with clustering validation, referencing the reliance on expert judgement, shortcomings of relative validity approaches, and its frequent focus on globular clusters. They introduced a metric to evaluate density-based clustering approaches which considers the density and shape of clusters – Density Based Clustering Validation (DBCV). It is built upon the premise that good density-based clustering solutions are governed by the lowest density regions within clusters in comparison to outside of it. Such regions are identified through the development of minimum spanning trees within each cluster, built upon symmetric reachability distances of points, and determined by the inverse of its density. The maximum distance between a pair of points in a cluster is a representation of the density sparseness while the minimum distance between a pair of clusters symbolise the density separation. DBCV compares the density sparseness against the density separation, whereby highly positive values (capped by +1) signifies representations with compact and separated clusters. Conversely, highly negative values (capped by -1) signifies representations with sparse and nearby (or even connected) clusters.

2.7. Binary Classification

Quinlan (1986) simplified a classification task to a series of arguments, packaged as a decision tree. They illustrate how class labels are represented by leaves and branches form a set of possible outcomes - tracing from the start of the tree (root), branches are taken in series (in accordance to the argument)

down to a leaf node whereby a final classification label is outputted. Through a greedy search algorithm (i.e. taking the best option with no considerations to the future), optimal splits over a mutually exclusive set of values of a variable are identified and the top-down approach is repeated in a recursive manner until all samples have been classified. They hinted at the potential challenges surrounding the generalisability of decision trees over an unseen dataset; mentioning considerations (e.g. pruning) and preferences towards simpler trees. In reference to it, they presented two sets of decision trees: a simple and complex one – highlighting risks of overfitting from the latter. The simple and complex decision trees presented by Quinlan are shown in Figure 11 and Figure 12 respectively.

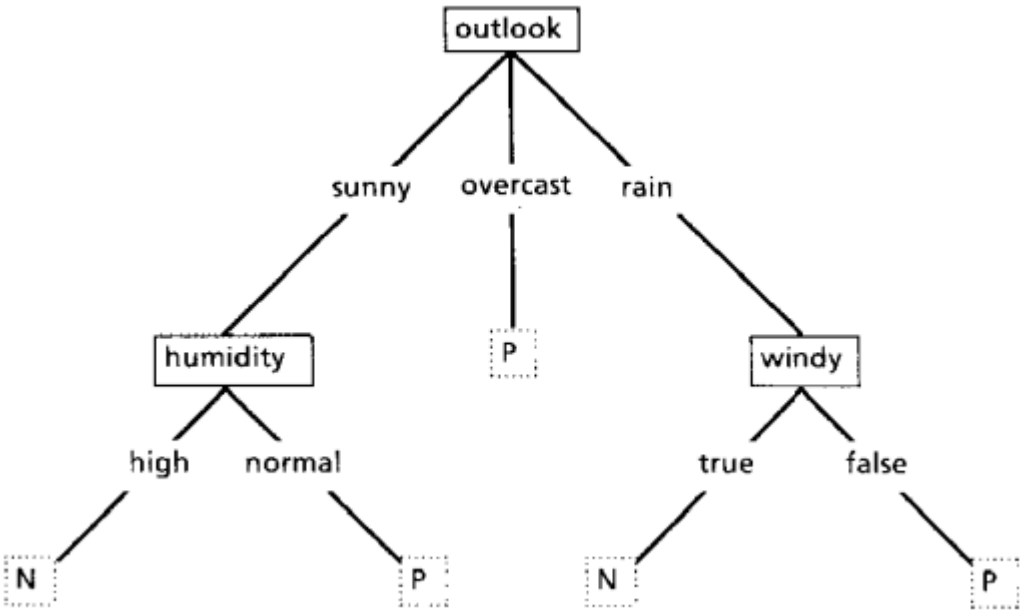


Figure 11: Simple decision tree (Quinlan, 1986)

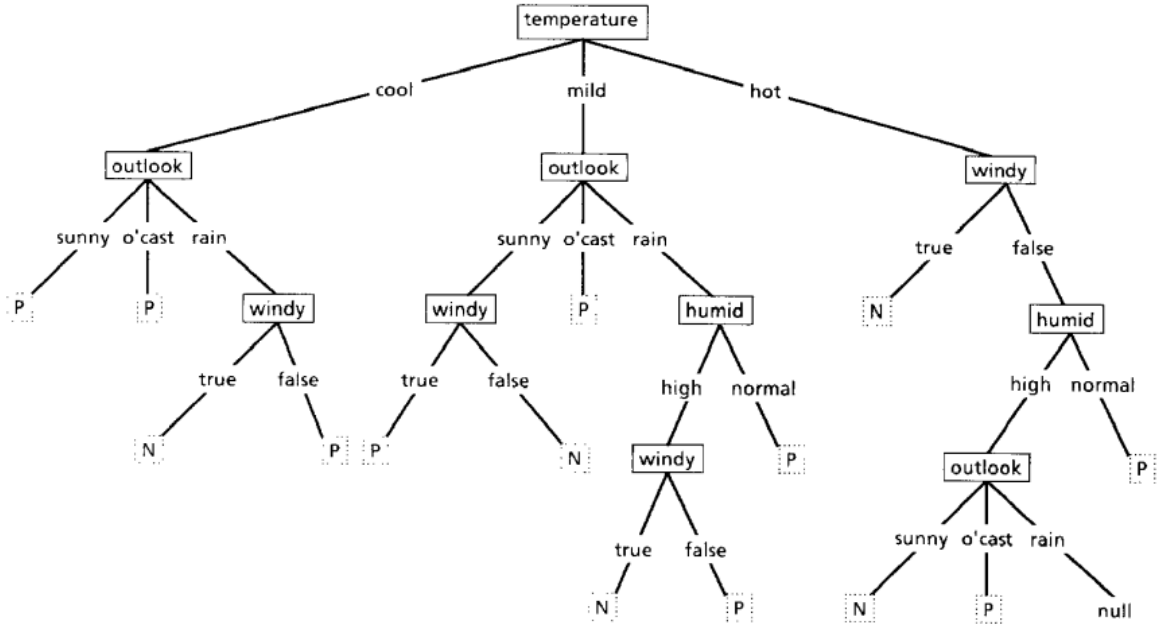


Figure 12: Complex decision tree (Quinlan, 1986)

Mitigating the challenges of a single decision tree, Breiman (2001) developed upon random forests – an ensemble of decision trees. As a bagging approach, random forests are made up of a deterministic number of decision trees fitted over different subsets of the training data (through random sampling) such that all trees are trained over a similar distribution. The samples excluded from the subset utilised for training were treated as 'out-of-bag' samples and used to determine the generalisability of the random forest by calculating the generalisation error. The collection of classifiers (trees) were applied onto the out-of-bag samples and predictions are obtained through majority voting or averaging, contingent on the task (e.g. classification or regression). They supplied an approach to measure variable importance through the orderly noising of each feature and measuring the percentage increase in misclassification error. They found key variables holding significant amounts of information but caveated the results with emphasis on understanding the features and pair-wise dependencies between them. Though a feature was perceived to be the second most important, it carried the same set of information as the first placed feature (i.e. high dependencies and correlation) and resulted in far smaller decreases in misclassification error as expected. Over a range of domains, they found random forests to perform as well as rivalling approaches like 'Adaboost' but with the added robustness towards noise.

2.8. Multi-label Classification

The practice of understanding text and language was summarised by Khurana *et al.* (2022) as Natural Language Understanding (NLU), a sub-domain of NLP. Approaches like Recurrent Neural Network (RNN) were adapted to NLP tasks like text classification by Liu, Qiu, and Huang (2016) but ran into challenges surrounding scalability due to the sequential processing of words and thus, proving parallelisation techniques difficult. As an improvement to the performance and computational

complexity, Devlin *et al.* (2019) introduced BERT; built upon the transformer architecture innovated by Vaswani *et al.* (2017).

The transformer model architecture is constructed from encoders, decoders, and three main innovations: positional encoding, attention, and self-attention. An illustrated representation is shown in Figure 13.

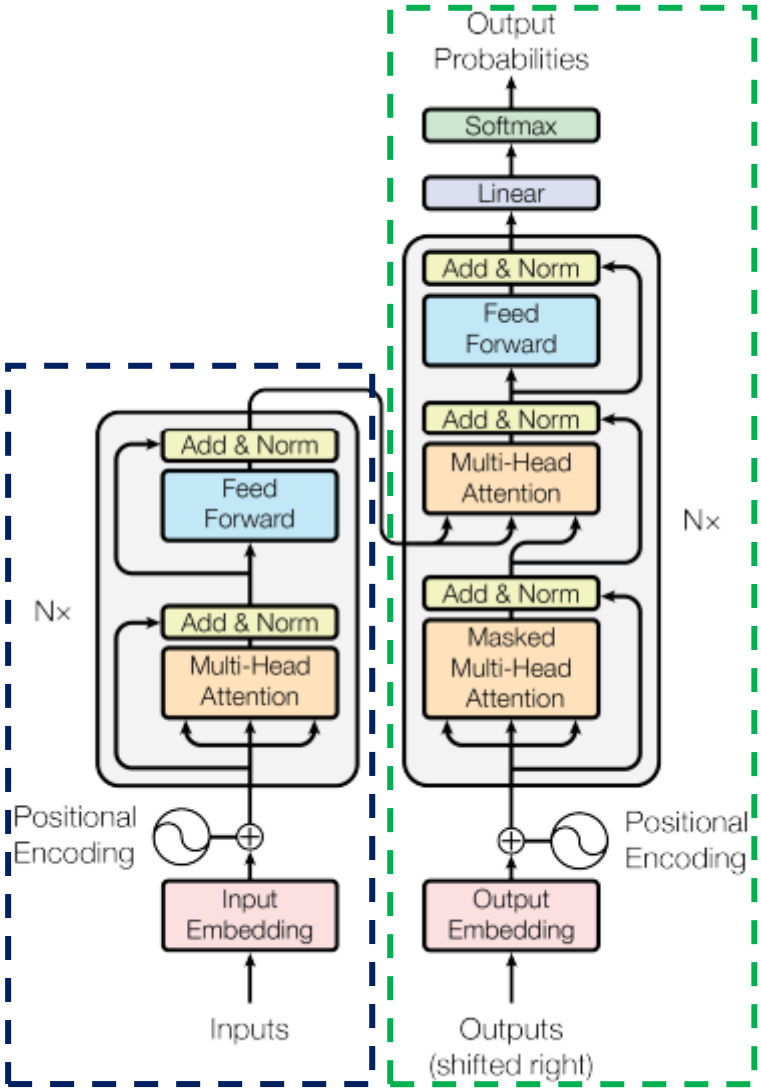


Figure 13: Transformer model architecture (Vaswani et al., 2017)

The components encased within the blue and green dotted lines represent the encoder and decoder respectively. The encoder takes words simultaneously and generates embeddings for each word while the decoder takes previously generated words and outputs the next word. The former learns about the language (e.g. grammar) while the latter learns about dependencies between words (e.g. mapping English to French words in a text translation setting).

Utilising a series of encoders, the BERT model architecture is formed with applications in sentiment analysis, text summarisation, and text classification (Devlin *et al*, 2019). They summarised the training of BERT models into two main steps: pre-training to understand language and fine-tuning over a specific task or domain. The training process is illustrated in Figure 14.

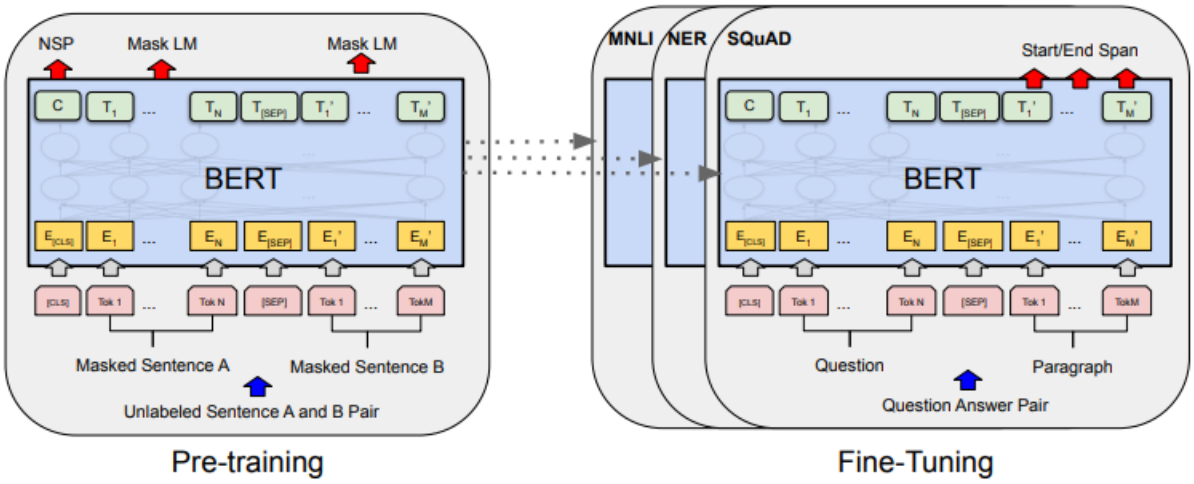


Figure 14: BERT training process – pre-training and fine-tuning (natural language inference, named entity recognition, and question and answering as examples) (Devlin et al, 2019)

In pre-training, BERT learns through simultaneous tasks of Masked Language Modelling (MLM or Mask LM) and Next Sentence Prediction (NSP). With MLM, a selection of words is randomly masked and the masked tokens generated are evaluated (e.g. the appropriateness of a masked token in the given setting). Conversely, NSP assesses whether the temporal order of sentences make logical sense (i.e. if it's contextually correct for one sentence to succeed another). Visually, from Figure 14, input sentences are converted to embeddings and the binary output C, represents results for NSP. Output word vectors T, correspond to input word vectors and are converted into a distribution through a softmax function. One-hot encoded vectors act as labels (i.e. the masked word is un-masked, providing the true label) and are compared to the softmax distribution via the Cross-Entropy Loss. During training, the cross-entropy loss is minimised, with focus on the loss resulting from the masked word (i.e. ensuring masked words are predicted correctly). The embeddings to the input take into consideration the pre-trained embeddings (token embeddings), sentence number (segment embeddings), and word number within the sentence (position embeddings). These embeddings are summarised in Figure 15. 'CLS' and 'SEP' are special tokens to represent the start of every input and separation between inputs (e.g. between two sentences) respectively.

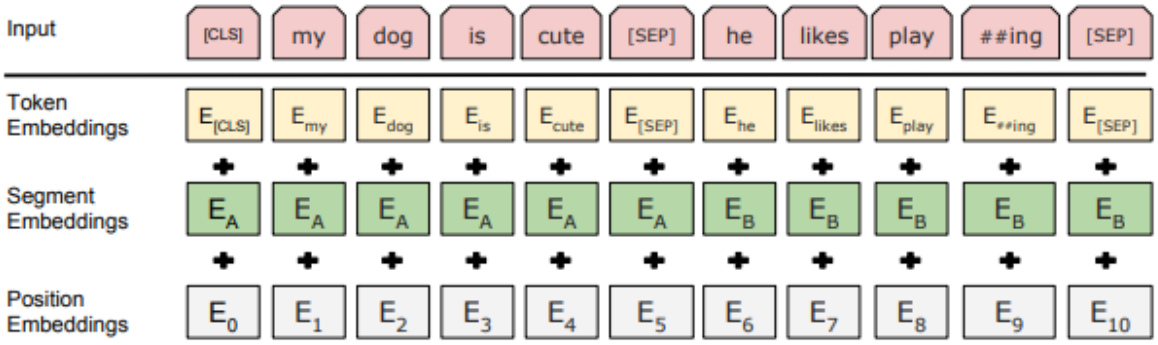


Figure 15: BERT input embeddings (Devlin et al, 2019)

During fine-tuning, the model is trained and the architecture is adapted to the specific task by adding additional layers (hidden and output) to the corresponding output. Figure 14 provided an example to a question/answer task where the output of the BERT model is altered to output starting and ending word which encapsulates the answer. Here, an output layer (with a number of preceding hidden layers) with two nodes provide such an approach.

Over a range of NLU tasks and question/answer datasets, Devlin *et al*, (2019) found BERT models to consistently outperform its counterparts like OpenAI's Generative Pre-trained Transformer (GPT).

2.9. Application to Study

The legality and ethics of web-scraping has been an on-going topic of debate and often falls within a grey area (Krotov and Silva, 2018). Concerns surrounding purpose of web-scraping, infringement on individual privacy, and damage to websites, form arguments against web-scraping. Since this project aims to base its study on web scraped reviews, ethics surrounding it needs to be a point of discussion. However, given that our study aims to prove such approaches are possible in the realm of company monitoring and serves as an academic exercise (rather than marketing it as a product), we believe that ethics are preserved. Adaptations to time intervals between scrapes also provides an option to alleviate the burden on web servers.

Since customer reviews behave similarly to tweets (e.g. noise and language used), processing steps (e.g. stop word removal and tokenisation) presented by Sriram *et al.* (2010) will be considered alongside added consideration on the specifics of reviews. Utilising special tokens (Howard and Gugger, 2020) provides a good alternative approach to data pre-processing but may be contingent on the choice of word embedding approach. Back-translation (Edunov *et al.*, 2018) supplies a way to generate synthetic data points in the presence of lacking data points or preferences for added noise in our dataset.

Word embeddings may prove a more appropriate approach in the setting of this study – using real-world reviews may accentuate challenges with sparsity within BOW approaches, alongside the loss of semantics and ordering of words within sentences (Mikolov *et al.*, 2013). GloVe (Pennington, Socher and Manning, 2014), word2vec (Mikolov *et al.*, 2013), and universal sentence encoders (Cer *et al.*,

2018) offer state-of-the-art approaches to word embeddings and may be more fitting within the context of this study. All 3 approaches shall be considered but universal sentence encoders have proven novel performances despite smaller datasets.

Dimensionality reduction techniques has justified its use for visualisation and evaluation on the quality of embeddings and clustering. PCA (Jolliffe, 2002) allow for a convenient approach to dimensionality reduction but is caveated by its linear approach and focus on separating dissimilar words apart versus inching similar words together (van der Maaten and Hinton, 2008). While t-SNE (van der Maaten and Hinton, 2008) patches the limitations surrounding PCA through a non-linear approach, it runs into difficulties with scalability and preservation of global structure (McInnes, Healy and Melville, 2020). In this regard, UMAP (McInnes, Healy and Melville, 2020) extends a more robust (and less computationally taxing) approach for dimensionality reduction within this study.

Clustering techniques have demonstrated its importance in unsupervised learning techniques as a way to uncover underlying patterns within the data. Though k-means clustering (Macqueen, 1967) grants a simple and intuitive approach to clustering, initial assignment of centroids require repeated experiments to mitigate this dependency. Rather, HDBSCAN (Campello *et al.*, 2013) extends a hierarchical and density-based approach to clustering – supplying a robust approach which outputs the most stable clusters. A hierarchical approach is also more suitable within the context of this study. Topics are convoluted and are unlikely to be independent of one another. Even niche and specific topics are often housed by a larger and broader topic – turnover times and live chat experience both being aspects of customer service for example. A hierarchical approach extends opportunities for grouping of clusters.

Within this study, topic modelling allows for topics to be assigned to clustered embeddings. LSA (Landauer, Foltz and Laham, 1998) and NMF (Lee and Seung, 2000) furnishes intuitive approaches to topic modelling through variations of SVD. LDA provides a probabilistic approach to topic modelling and outputs explicit sets of probabilities. Regardless, all three approaches are built upon the assumption that documents are made up of a number of topics and topics are made up of words which are commonly used together.

The evaluation metrics covered furnishes ways to assess model performance over supervised, semi-supervised and unsupervised learning tasks. External and internal measures often require a set of ground truth labels and are often derived from the same observations used respectively (Dudoit and Fridyland, 2002). External metrics like accuracy and f-score offer different lenses to evaluate supervised and semi-supervised learning techniques. Conversely, internal metrics like DBCV (Moulavi *et al.*, 2014) presents approaches to assess the quality of clustering.

A multi-labelled dataset for multi-label classification is looked to be produced and a series of binary classifiers administers such an approach. Random forest presents an approach, built upon an ensemble of decision trees (Quinlan, 1986), for binary classification. By controlling the number of

features considered at each split, tree depth, and number of trees, random forests mitigate the overfitting risk commonly found in decision trees (Breiman, 2001). Using the embeddings as inputs and clusters as labels, we can leverage upon random forests to build a multi-labelled dataset.

BERT (Devlin *et al.*, 2019) innovated novel results in text classification (and other NLP-related tasks) since the introduction of the transformer architecture (Vaswani *et al.*, 2017). Built upon a series of encoders, BERT models can be fine-tuned to our multi-label text classification task through transfer learning and adaptation of its architecture (e.g. adding on hidden and output layers to produce classification predictions).

3. Methods

As stated in Section 1, this research project aimed to draw insights on a company's performance from customer reviews using NLP techniques. To do so, we've designed and implemented an end-to-end NLP pipeline, as laid out in Figure 16, which employed unsupervised learning techniques over an unlabelled dataset before transitioning over to supervised learning on a self-assigned labelled dataset, with semi-supervised learning serving as a bridge between the two.

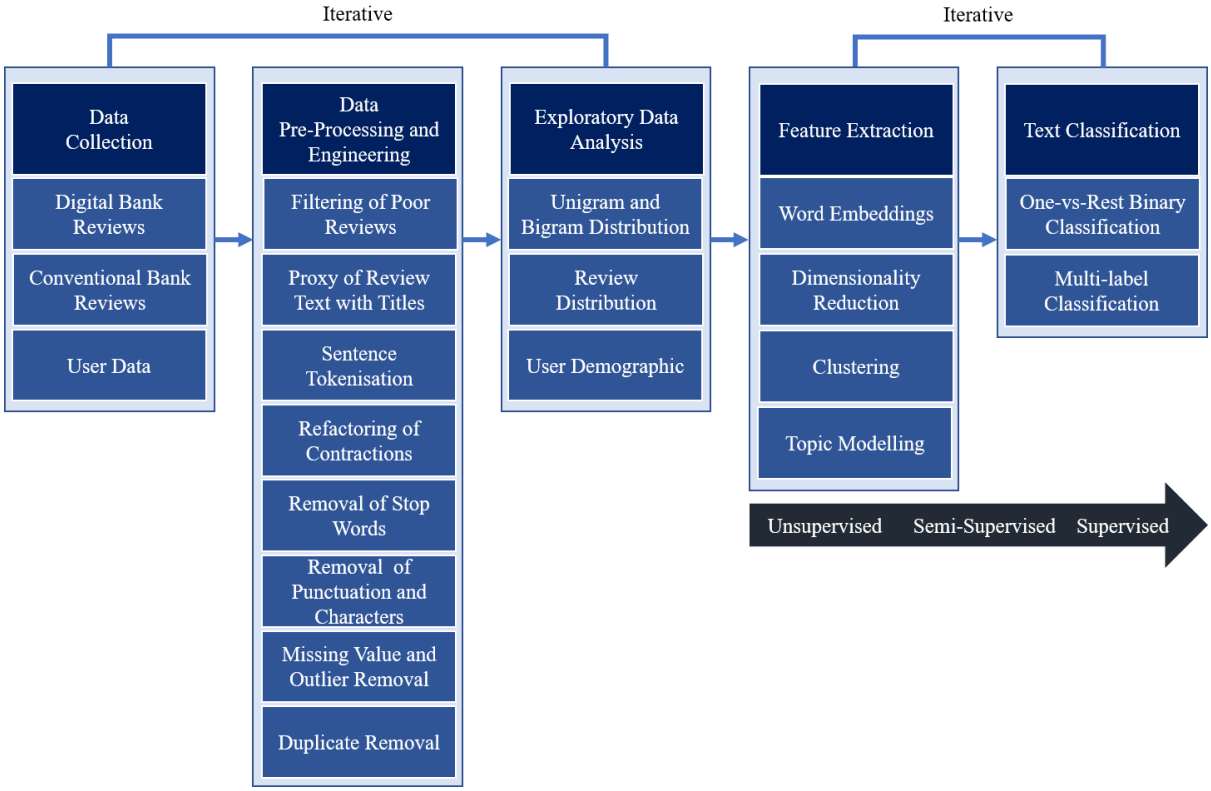


Figure 16: End-to-end NLP pipeline

Working from data collection to text classification, the following subsections discusses each component of the pipeline to further detail. Each subsection highlights the: component goals, implementation details, library dependencies, and iterative process. The components and techniques throughout the pipeline were scaled through virtual machines (VM) hosted by Amazon Web Services (AWS) to account for computational and resource intensities. Appendix 9.1 provides a summary of resource dependencies throughout the pipeline.

3.1. Data Collection

In data collection, we looked to create a text-based dataset of English public customer reviews within the banking and financial sector. The dataset was comprised of a combination of reviews from several banks which have operations and presence within the UK - both conventional (i.e. Lloyds, HSBC, Barclays, and Santander) and digital challenger ones (i.e. Monzo, Revolut, Wise, Starling Bank, and

N26). Using an open-sourced library called 'Scrapy', we curated datasets sourced and scraped from Trustpilot - an online platform for individuals to post reviews of companies based on their experience.

Scrapy is an open-source web-crawling framework in Python whose architecture utilises individual self-contained crawlers, 'spiders'. Spiders are assigned settings and are given a set of instructions to crawl which dictates the information that will be scraped (scrapy/scrapy, 2022). In the context of this study, these set of instructions take the form of specified Extensible Markup Language Paths (XPaths) which determines the location of key features of reviews (e.g. review date) left on TrustPilot webpages for each specified bank that were ultimately scraped[1].

With this approach and employing a Virtual Private Network (VPN) we curated two types of datasets in total, namely: one which is review-centric (review dataset) and provides context to the reviews and another which is user-centric (user dataset) and provides context to the users who left the reviews. Figure 17 and Figure 18 displays the key features to the review and user dataset respectively.

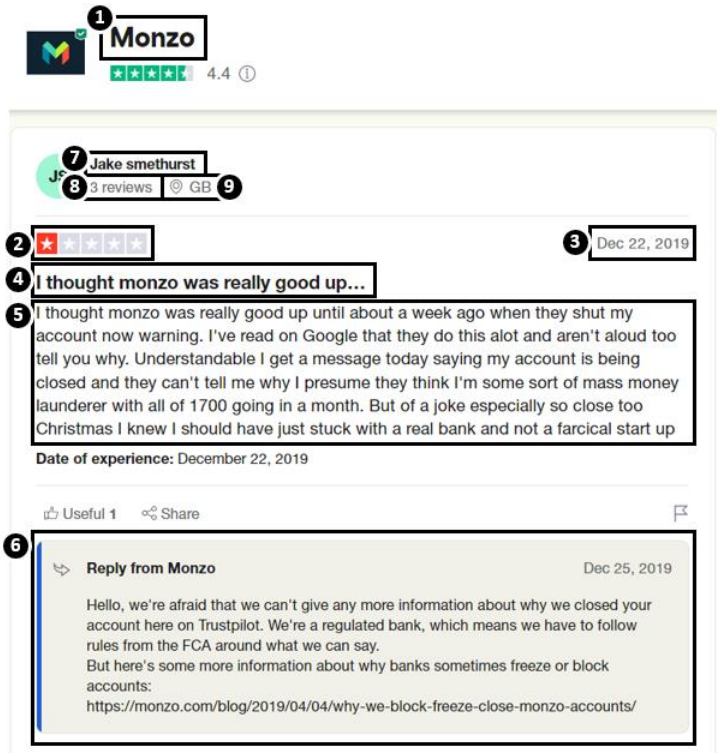


Figure 17: Key components in the review dataset

To form our review dataset, draw insights and gain context, as will be discussed in Section 2.3, we identified nine key features. Specifically:

1. Company: The company which the user is referring to
2. Review rating: A rating given by the user ranging from 1 to 5 stars, reflecting a poor and excellent experience respectively

[1] The web-scraping settings used are listed in Appendix 9.2

3. Review date: The date the review was posted
4. Review title: A title to summarise or indicate the subject of the review
5. Review text: An explanation and context of a user's experience
6. Company reply: A message, posted by the company, in reply to a user's review
7. User Uniform Resource Locator (User URL): The URL link to the user's profile
8. Number of reviews: The total number of reviews left by the user onto companies
9. Review location: Location whereby the review was posted from

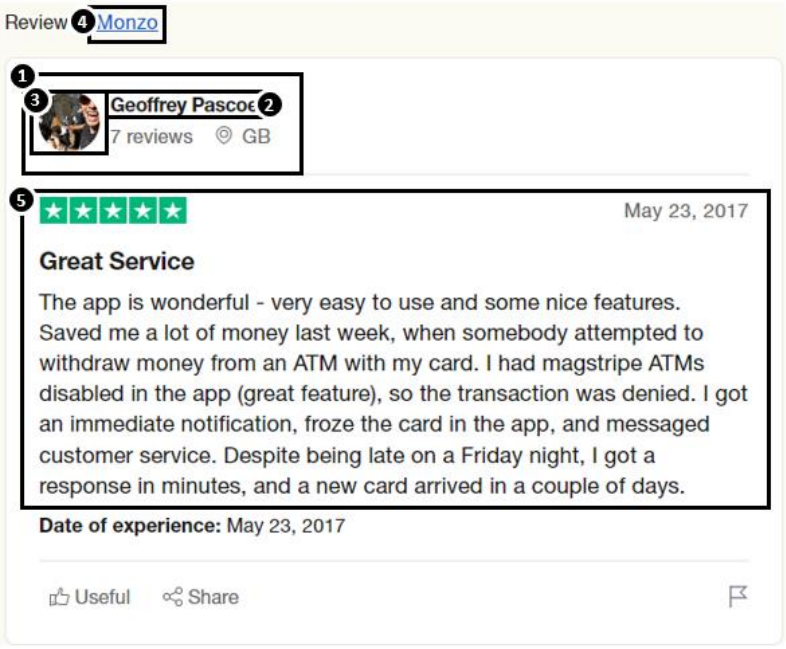


Figure 18: Key features in the user dataset

To form our user dataset, we identified similar key features to the review dataset, though with some additional features pertaining to the user. In particular:

1. User URL: The URL link to the user's profile
2. Username: The user's name which they go by on TrustPilot
3. User display image: The user's profile picture
4. Company: The company which the user is referring to
5. Review details: Similar features to the review dataset (i.e. review rating, date, title and text)

Despite the similarities, the user dataset differs from the review dataset as all reviews left by the user were scraped and recorded - including reviews of companies outside the banking or financial sector.

The web-scraping process was divided into batches to account for the number of the reviews, users, and any interim processing or exploratory data analysis (EDA), resulting in marginal mismatches in data collection dates. Table 1 summarises the datasets and their respective collection dates.

Dataset	Collection Date

	Review Dataset	User Dataset
Challenger Banks excluding N26 (Monzo, Revolut, Wise, and Starling Bank)	10/06/2022	22/06/2022
N26	22/06/2022	22/06/2022
Conventional Banks (Lloyds, HSBC, Barclays, and Santander)	23/06/2022	23/06/2022

Table 1: Data collection dates

The scraped data will be stored as a combination of Comma-separated Values (csv) and JavaScript Object Notation (json) objects, depending on the size and complexity of the dataset. Csv files allow for easier ad-hoc visualisation but run into limitations with the number of characters stored per cell.

3.2. Data Processing and Engineering

This section looked to process and transform the original review dataset into a cleaned dataset comprised of reviews' sentences. From the perspective of a company, poorly rated reviews serve as better indicators for prioritisation, improvements and innovation. For example, frequent complaints about account openings and praises about customer service suggest that account opening procedures should be prioritised and improved upon over customer service. Consequently, only poorly rated reviews were kept from the review dataset for further processing down the pipeline. Poor and good reviews were defined as reviews with ratings of between 1 to 3 and 4 to 5 respectively.

By taking the poor reviews, proxying any review text with its review titles (given it only had a title), and splitting or tokenising its reviews into sentences (sentence tokenisation), a review sentence dataset is produced and subsequently cleaned through the refactoring of contractions into its full forms2 and removal of common occurring words3 which carry little information (stop words). Further cleaning is performed through the removal of certain punctuations, characters4, duplicated sentences, sentences with missing dates, and sentences deemed as outliers. Outliers were defined as sentences with fewer than three words.

The reviews were tokenised into its sentences using an open-sourced NLP library, 'spaCy'. SpaCy offers transformer-based pipelines trained over written text (i.e. blogs, news, and comments), allowing for it to be loaded and employed, rather than training one from scratch, and making it suitable to the use case of this study (given the similar nature of reviews). The 'en_core_web_trf' (*spacy/en_core_web_trf*, no date) pipeline was employed in particular.

[2] List of contractions and respective full forms is listed in Appendix 9.3
[3] List of stop words is listed in the Appendix 9.4
[4] List of punctuations and characters is listed in the Appendix 9.5

Bank names were treated as stop words and removed from its respective reviews after iterative EDA on the distribution of most common unigrams and bigrams. A bank's review would frequently reference its name (e.g. Monzo reviews would frequently reference Monzo) to provide context to the subject of the review. However, the name of the company is recorded, as seen in Section 3.1, and serves as an indicator to the subject of the review instead. Therefore, bank names in its own respective reviews carry little information and were treated as stop words. On the other hand, bank names (aside from its own) referenced in other bank reviews provide insights on competition. For example, poorly rated Revolut reviews that frequently reference Monzo may suggest that part of Revolut's services are outperformed by Monzo.

Other common processing techniques like stemming and lemmatisation were not performed due to potentially altering the semantic meaning, word embeddings, and difficulty for manual evaluation. End-of-sentence punctuation was not removed to prevent inaccurate word embeddings and capture sentiment within the sentence (e.g. a sentence ending with an exclamation mark could suggest higher levels of anger or happiness as compared to a sentence ending with a period). Numerical characters were also not removed to ensure complaints surrounding timeframes are recorded – a common element of complaints surrounding customer service for example.

Figure 19 illustrates the data processing and engineering flow through an example review. The reviews consist of text and need not be proxied by their titles. Each review is tokenised into its respective sentences and contractions like "wasn't" within the sentences are refactored to its full-form: "was not". Stop words like "the" and "I" are removed along with converting the sentences to lower case, punctuation like commas are removed while end-of-sentence punctuation marks (e.g. periods and exclamation marks) are kept. The sentence "terrible experience!" is dropped as it tagged as an outlier while an instance of "not easy open account." is dropped due to multiple instances (i.e. duplicates).

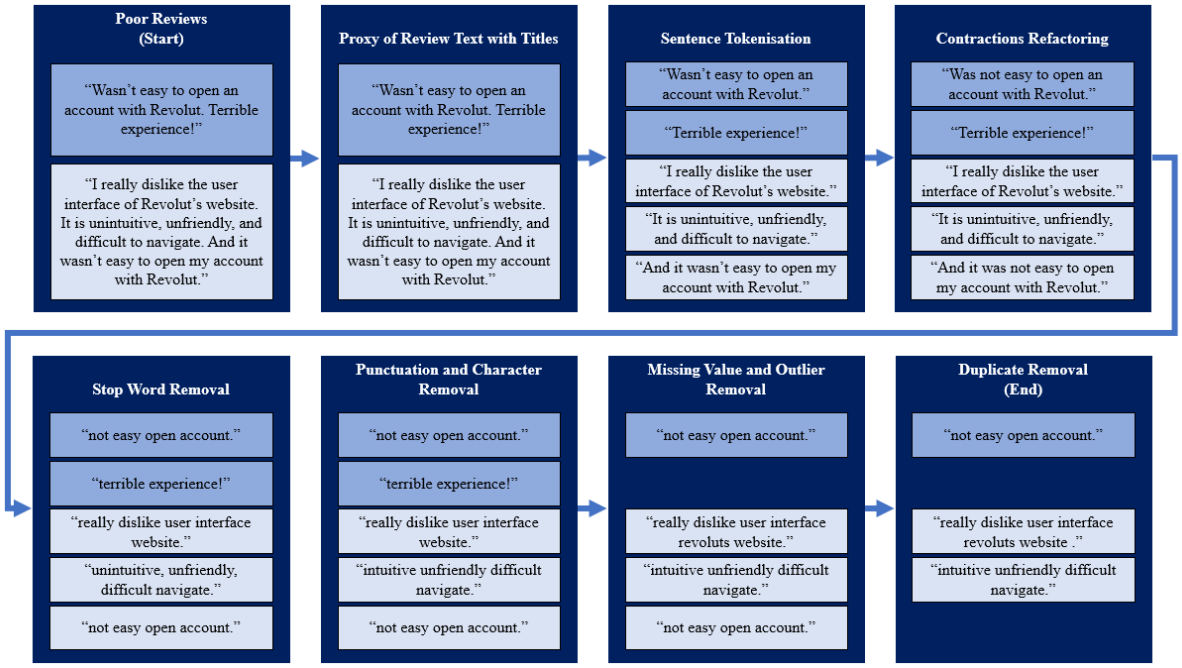


Figure 19: Data processing and engineering example

3.3. Exploratory Data Analysis

EDA served as an exercise to identify any preliminary trends that exist within the dataset, improvements to prior components within the pipeline, and considerations to make further down the pipeline; resulting in an iterative process with intermediate results. This included considering additional features to collect, data cleaning steps to take, and whether separate models should be developed for challenger and conventional banks. The EDA were grouped into three main categories, namely: user demographic, review, and unigram and bigram distribution. 'Pandas' and 'matplotlib', Python libraries for data manipulation and plotting respectively, were utilised for various data aggregation, transformation, and visualisation techniques.

The distribution of reviews visualized the variation in online presence between the types of banks (challenger and conventional) and structural distinctions between types of reviews (good and poor). This involved plots to summarise and assess the number of reviews, reply behaviour, and review length holistically and over time.

Comparing the user demographic (i.e. user age, gender and location) determined any asymmetry in users between types of bank. While the location of the user is provided, as seen in Section 3.1, the gender and age needed to be inferred through a combination of the user's name and profile photo. For example, an open-source library like 'gender-guesser', allowed for a gender to be derived by using the user's name as an input.

Iteratively evaluating the distribution of common unigram and bigrams aided the data cleaning process by finalising the steps to take within it and a stop word list as seen in Section 3.2. Between

banks and reviews, preliminary topics identified may differ due to differences in product offerings (e.g. cross currency transfers for Wise and savings account for HSBC) and quality of services or offerings (e.g. account closure being simple while account opening being difficult) respectively.

3.4. Feature Extraction

Feature extraction aimed to generate numerical features whilst preserving any underlying information that exist within the dataset (e.g. semantics). By employing unsupervised learning methods, this section sought to create a labelled dataset comprised of review sentences, labels, and its corresponding topics, for classification model development further down the pipeline. For example, a review which quotes "customer service is slow and unresponsive" will be tagged as "customer service" and labelled as label 7 – each topic will be assigned to a numbered label. A subset of review sentences was first used as a proof-of-concept to assess the feasibility of the series of techniques within this section. After this was deemed successful, the series of techniques were scaled up onto a larger subset of review sentences.

With textual data in the form of review sentences, transformations onto the numerical space are required prior to further processing. Word embeddings transforms sentences into vector representations which preserves the meaning of words and sentence. Words which are closer together in the vector space indicate similarity and relationships between words are also preserved (e.g. relationships between male-female and king-queen). Reviews were tokenised into sentences in Section 3.2 to prevent the dilution of word embeddings – from convoluted reviews with multiple sentiments for example. By utilising pre-trained sentence encoding models (universal sentence encoder) found on TensorFlow Hub, a 512-dimension vector representation was obtained for each review sentence.

A challenge that arises from a 512-dimension vector is that information is not distributed uniformly across dimensions. More information is carried in certain dimensions over others resulting in an infeasible iterative task of evaluating large numbers of pair-wise (or tripartite-wise) combinations when determining the dimensions which carry the most information. By implementing dimensionality reduction techniques, specifically UMAP, the 512-dimension vector is reduced down to an explicit number of dimensions and tuned over 'n_neighbors' and 'min_dist' hyperparameter values. The former reflects preferences in local over global structures that exist within the data, while the latter determines how densely packed points are – affecting the grouping of embeddings and topological structure. 3-dimensions were selected to allow for visual evaluation resulting in a reduced 3-dimensional vector representation of the review sentences.

With a 3-dimensional vector representation of the review sentences, clustering techniques were employed to group review sentences into clusters according to underlying information (e.g. semantic meaning across sentences). HDBSCAN was identified as a suitable approach to group and label the review sentences. The non-parametric property of HDBSCAN prevents the need for the number of clusters to be pre-defined, allowing for potential clusters initially not considered to be discovered A

hierarchical approach, instead of a flat approach, was deemed to be more appropriate since it accounts for overarching clusters (and topics) that can be split into smaller individual clusters (and topics). For example, a large cluster associated to an overarching topic like customer service may house smaller clusters associated to more specific topics like turnover times and politeness.

The 'min_cluster_size' hyperparameter was tuned which sets the minimum number of points required to be constituted as a cluster. Given that the quality of the clustering is contingent on, not only, min_cluster_size but also the 3-dimenisonal vector representation (and thus, n_neighbors and min_dist), the three hyperparameters were tuned in combination and evaluated through visualisation and the relative validity of each hyperparameter combination; resulting in a 3-dimensional vector representation of the review sentences with assigned cluster labels. Table 2 lists the hyperparameter values which were tuned over.

Method	Hyperparameter	Values
UMAP	n_neighbours	[15, 50, 100]
	min_dist	[0.0, 0.5, 0.99]
HDBSCAN	min_cluster_size	[15, 30, 60]

Table 2: Hyperparameter values for UMAP and HDBSCAN

From the assigned clusters, a potential point of consideration is the overlap or similarities of clusters and non-sensical clusters from the view of this study. With the former, multiple clusters may depict a similar topic (e.g. five clusters surrounding customer service). The latter, for instance, may deal with clusters which surround a sentiment instead of a topic. For example, a group of reviews which mentions a bad experience without elaborating on the service or offering provides no additional information since the reviews used are all poor reviews. Contrastingly, a group of reviews which highlights the particular service or offering they've had a negative experience with, provides more insights. Regardless, to gain an understanding of the clusters, topic modelling techniques like LDA and NMF were used to assign topics to each cluster with minimal manual intervention. From it, the top topics of each cluster are used as a depiction of the given cluster and manual assessments of grouping similar clusters and/or reassigning non-sensical clusters as noise were made; resulting in a labelled dataset comprised of review sentences (and its corresponding embeddings), labels, and topics.

Figure 20 illustrates an example to the feature extraction process. The cleaned and processed review sentences from Section 3.2 are transformed into word embeddings – a 512-dimension vector representation. Upon hyperparameter tuning for dimensionality reduction and clustering, a reduced 3-dimensional vector representation with assigned cluster labels is produced. LDA and NMF assigns topics to each cluster and manual evaluation allows for regrouping and reassigning of clusters. For example, clusters 1 and 2 both surround the account opening process and thus was grouped to form a

single cluster. Cluster 3 only provides a sentiment without elaboration on the service or offering (information that is inherently deduced from the review rating) and was reassigned to the noise cluster.

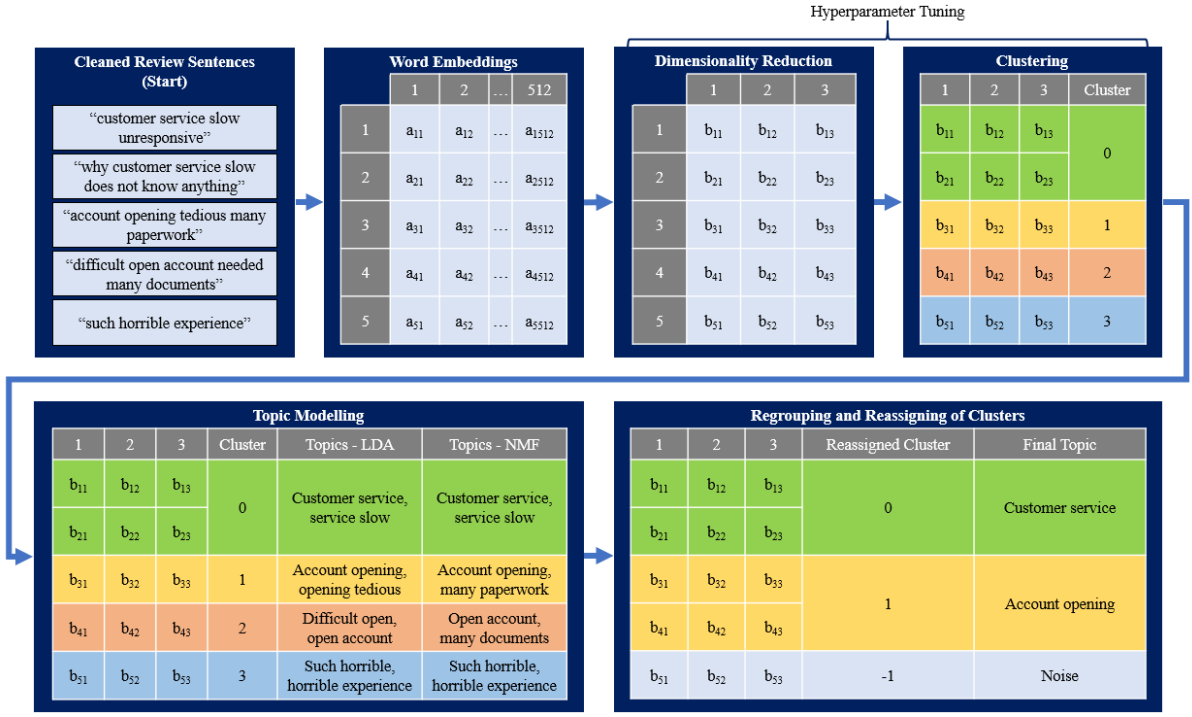


Figure 20: Feature extraction example

3.5. Text Classification

This section looked to develop a multi-label text classification model which classifies text to a topic (or series of topics) given the word embeddings. By utilising semi-supervised learning methods, this section sought to first develop a series of binary classifiers over a subset of the data using the word embeddings as input and the clusters from Section 3.4 as the true labels. After iterative evaluation using external performance metrics, investigating false positives and false negatives, and further regrouping of clusters (if necessary), these series of binary classifiers are to be applied over the entire dataset – forming the true multi-label labels for further model development. With a fully labelled dataset, supervised learning techniques were implemented to train a multi-label topic classifier model and conclude experiments of this study after evaluating its performance.

Streaming the outputs from Section 3.4 as the inputs to this section, a series of random forests are developed – one for each cluster label. A random forest is a collection of decision trees used for (within this study) binary classification and covers the overfitting limitations found in decision trees, though at the expense of time complexities (Breiman, 2001). By tuning the 'n_estimators', 'max_depth' and 'max_features' hyperparameters, the number of decision trees. depth of the decision tree and number of features to consider at each node split can be controlled for; mitigating the risk of overfitting. Using 80% of the subset of data for training (and validation), all 512 dimensions of the word embeddings as input, clusters as true labels, tuning over the hyperparameters using grid search, and repeating it for

each cluster, a series of binary random forest classifiers (one-vs-rest random forest) are produced. Samples from the noise cluster is down-sampled to the average cluster and subsequently, samples which are not from the given label are down-sampled further to the number of samples in the given label (i.e. forming a training set whereby samples from the label makes up 50% of it). The latter mitigates biases within predictions while the former accounts for skewness in the distribution of the training set. Grid search was performed using 'GridSearchCV' and selects the optimal hyperparameter combination with the highest training accuracy. Table 3 lists the hyperparameter values which were tuned over.

Hyperparameter	Values
n_estimators	[1000, 5000, 10000]
max_features	[25, 50, 100]
max_depth	[100, 500, 1000]

Table 3: Hyperparameter values for random forest

After determining the optimal hyperparameter combination, the random forest is retrained using k-fold cross validation (over 10 folds); aggregating the predictions over the training and validation sets, and selecting model parameters from the fold with the highest validation accuracy. Given that a datapoint is included across multiple folds, majority voting was adopted to get an aggregated classification prediction. For example, for a datapoint that exists within 9 folds, 9 sets of predictions are obtained from each random forest developed over each fold. An aggregated classification prediction is obtained by determining the majority classification label.

In addition to serving as inputs to the multi-label text classifier model development, the series of random forests provides an additional dimension when evaluating the quality of clustering and the need for manual reassignment or regrouping. Through a combination of false positives, false negatives and soft clustering probabilities from clustering in Section 3.4, frequent misclassification can be assessed and manual regrouping of clusters can be considered. For example, review sentences from the 'turnover time' cluster may be frequently misclassified under the 'customer service' cluster (i.e. false positive) and the soft clustering probability associated to sentences from the former may be close to 0.5 (i.e. indecisively assigned to the given cluster). Thus, it may be deemed as beneficial to group the 'turnover time' cluster with the 'customer service' cluster. Furthermore, it acts as an evaluation exercise on the performance of each model (along with performance metrics like accuracy and f1-score).

Applying the series of random forests over the remaining dataset, a multi-labelled dataset is obtained. Review sentences are potentially tagged as multiple topics (e.g. both, business account and account closure related) and function as the true labels to the multi-label dataset. Parsing 80% the cleaned sentences as inputs of the training and validation set, and a multi-labelled array as the true labels, transfer-learning is applied onto a pre-trained transformer-based classification model, BERT (Devlin *et al*, 2019), to develop a multi-label text classification model. Transfer learning takes upon

parameter values from the pre-trained model ('bert-base-uncased') whilst adding a hidden layer and final layer (for text classification) with a rectified linear unit ('ReLU') and sigmoid activation function respectively. Binary cross-entropy ('BCE') serves as the loss function to train over which takes the logarithm of classification probabilities to exponentially penalise wrong predictions – classification probabilities which fall further from the true label are penalised more than those which are closer. Weights are fine-tuned through backpropagation and the model is trained over 5 epochs, with early stopping whereby training is terminated in the presence of consecutive increases in validation losses. The final resulting model is a BERT multi-label topic classification model which takes word embeddings as inputs and classifies it to a topic or series of topics.

Figure 21 provides an example to the text classification process. Samples tagged as noise are down-sampled to the average sample size across all clusters. Developing a binary random forest classification model for cluster 0, samples with binary label tags of 0 (i.e. not from cluster 0) are down-sampled further to make up the same number of samples with binary label tags of 1. Hyperparameters of the random forest are tuned and the series of random forest are applied onto the cleaned text (and word embeddings) to form a multi-labelled dataset. Utilising these as inputs and leveraging on transfer learning over a pre-trained BERT model, a BERT model is trained to the context of this study assigns text with a topic label or series of topic labels.

Figure 21: Text classification example

4. Results

This section mirrors the format of Section 3 and presents the outputs from the outlined pipeline. Each subsection will present the results, commentary, findings and evaluation where necessary.

4.1. Data Collection

Table 4 summarises the number of reviews collected from each bank.

Bank	Number of Reviews
Lloyds	2645
HSBC	5975
Barclays	1292
NatWest	3764
Santander	4372
Revolut	91445
Monzo	21131
Wise	132663
Starling	28940
N26	6353
Total	**298580**

Table 4: Size of review dataset by bank

Though a similar distribution to Table 4 can be displayed for the user dataset, reviews onto multiple banks left by the same user resulted in a marginally smaller user dataset for certain banks. For example, a single user leaving a review onto Monzo, Wise and HSBC results in a review dataset of size three as compared to a user dataset of size one. This results in a user dataset comprised of 295774 users in total.

4.2. Data Processing and Engineering

Table 5 compiles the number of review sentences for each bank after each given processing step. Other processing steps discussed in Section 3.2 (e.g. contractions refactoring and stop word removal) affects the number of words in a given sentence rather than the number of review sentences. Conversely, the processing steps listed in Table 5 affects the number of review sentences rather than the number of words in a given sentence.

Bank	Number of Poor Reviews	Number of Review Sentences		
		Sentence Tokenisation	Missing Value and Outlier Removal	Duplicate Removal
Lloyds	2208	13295	12742	12664
HSBC	5533	32739	c31551	31151

Barclays	1143	5986	5689	5646
NatWest	3373	19988	19135	18947
Santander	3932	23378	22512	22175
Revolut	10562	60092	57730	57055
Monzo	2740	13044	12436	12307
Wise	9847	42354	40608	40160
Starling	4051	21571	20680	20516
N26	2184	13846	13199	13078
			Total	**233699**

Table 5: Number of review sentences after given processing steps

On average, around 4% and 1% of the review sentences are dropped after missing value and outlier, and duplicate removal respectively. A weighted average, weighted by the number of reviews, of the number of sentences per review suggest differences between poor reviews of challenger and conventional banks. Particularly, 5.14 and 5.89 sentences make up a single poor review on average for challenger and conventional banks respectively.

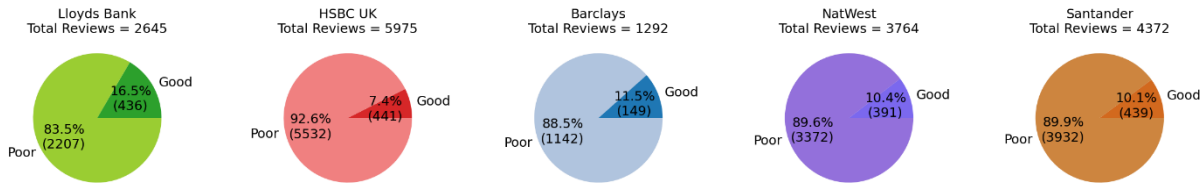
4.3. Exploratory Data Analysis

This subsection is divided into three further subsections for each EDA category as discussed in Section 3.3 – review distribution, user demographic, and unigram and bigram distribution.

When conducting EDA, we sought to understand the difference between challenger and conventional banks by looking at their review composition on TrustPilot. We also investigated evidence to suggest that conventional bank reviews are channelled out through formal physical platforms while challenger banks interact with online platforms. Review lengths were compared for good and poor reviews, in addition to demographic analysis across bank types and user locations. Analysis on the stop word selection process was also conducted, where descriptive diagrams were created.

4.3.1. Review Distribution

Figure 22 outlines the distribution of reviews by bank.



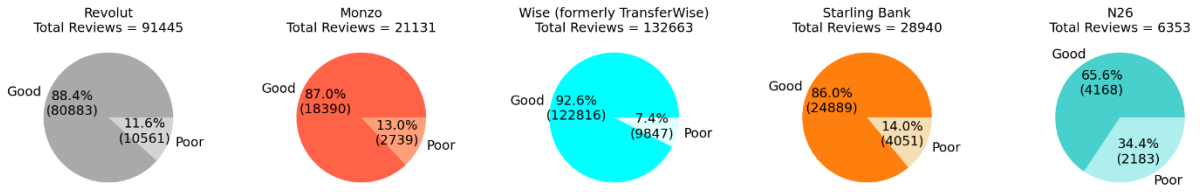


Figure 22: Distribution of reviews by bank

While good reviews make up the majority of reviews for challenger banks, the reviews of conventional banks are dominated by poor reviews. The largest percentage of good reviews amongst conventional banks is seen in Lloyds with 16.5% whilst the equivalent smallest percentage amongst challenger banks stand at 65.6% - N26. The number of reviews left on challenger banks is 15.54 times greater than that of conventional banks on average, showing dissimilarity in presence on TrustPilot. Though the latter could be alluded to difference in customer demographic, it could also be due to the difference in nature between bank types. While it can be assumed that both types of banks have formal pathways to submit reviews (e.g. on the respective websites), challenger banks may have a larger presence online due to not only its services and offerings being entirely online, but reviews being syphoned away through physical platforms for conventional banks (e.g. satisfactory surveys at physical branches) as well. Figure 23 reiterates the contrast in online presence of banks when plotting the percentage of reviews replied (i.e. number of replies divided by the total number of reviews for the given rating) by rating, from 1 to 5 stars.

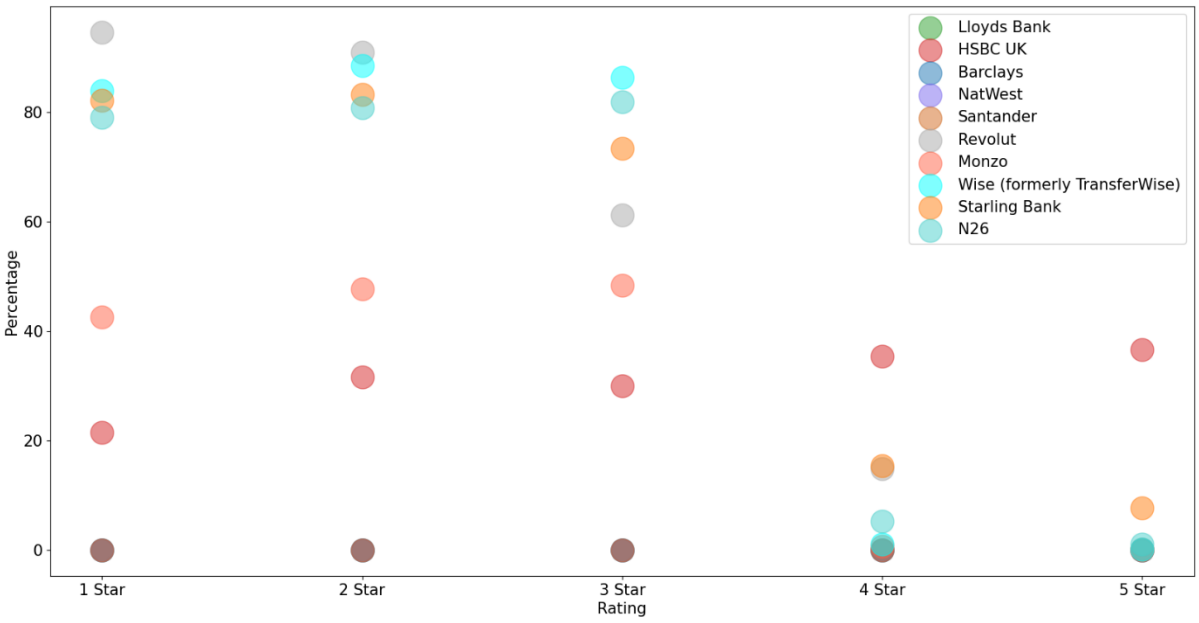


Figure 23: Distribution of company replies by rating

These results suggest that challenger banks show more focus on online public reviews with all challenger banks, excluding Monzo, replying to around 80% of poor reviews. Contrastingly, conventional banks excluding HSBC, provide no replies across all ratings. Monzo and HSBC behaves similarly for poor reviews by addressing similar amounts of reviews across the ratings (40 to 45% and

20 to 30% respectively) but deviates for good reviews – HSBC continues to reply to reviews while Monzo does not. Monzo aside, common behaviour amongst challenger banks are also seen with an inverse relationship between reply rates and rating (i.e. the lower the rating, the higher the reply rate). This outcome makes sense intuitively, as companies prioritising the resolution of complaints over acknowledging good experiences would be understandable

The idea of prioritisation is further supported when plotting the marginal distribution of review replies and supplementing it with the marginal distribution of reviews over time in Figure 24 and Figure 25 respectively. Both plots are normalised through min-max normalisation, allowing for trends to be analysed between banks despite the difference in magnitude. Min-max normalisation normalises the marginal number of replies (or reviews) at each month by the difference between maximum and minimum replies (or reviews) for each respective bank. Thus, 100% of replies at a given month represents the maximum number of replies the bank has issued over time, while 0% is conversely associated with the minimum.

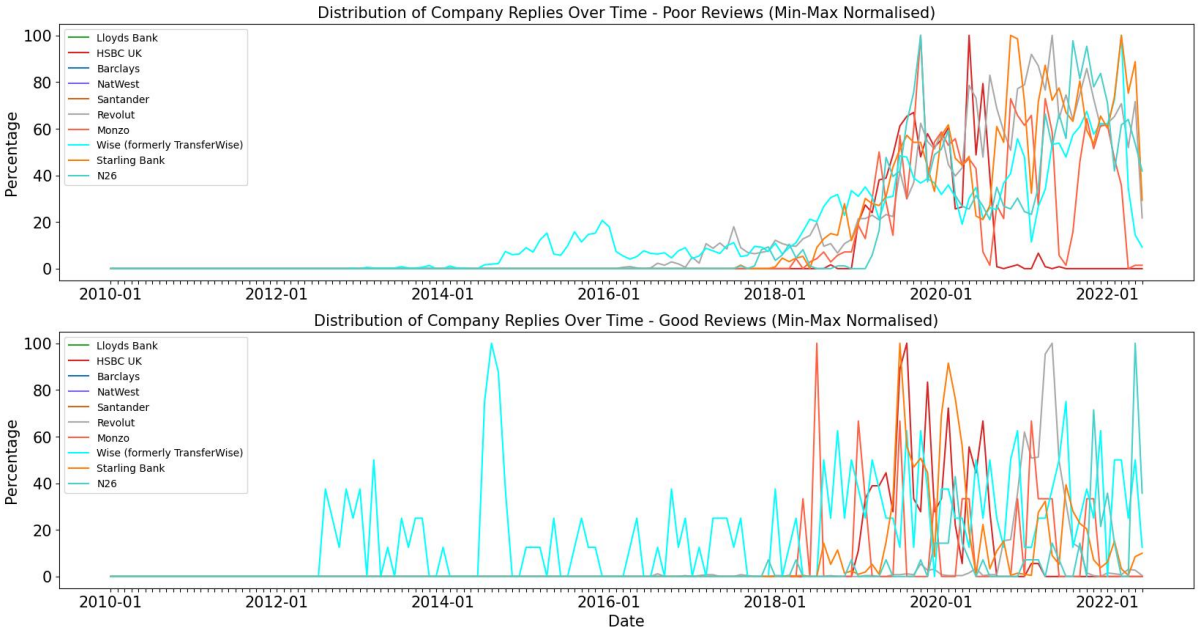


Figure 24: Marginal normalised distribution of replies over time by rating

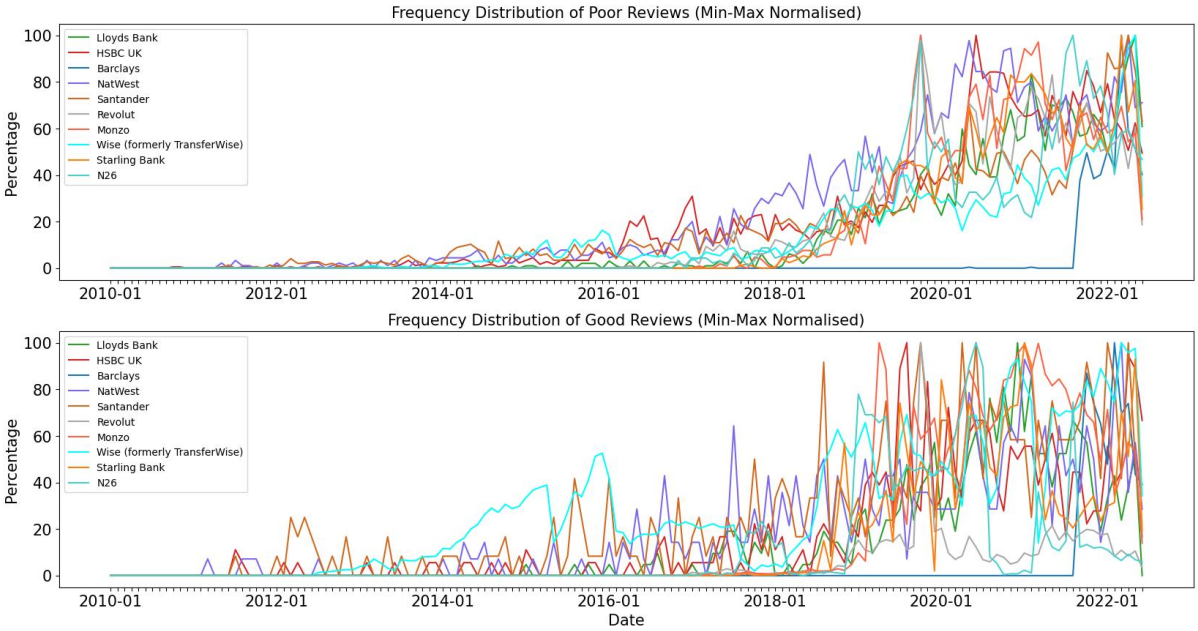


Figure 25: Marginal normalised distribution of number of reviews over time by rating

Taking Wise as an example, between 2013 to 2015, the reviews were mainly comprised of good reviews and correspondingly, the reply rates for good and poor reviews were high and low respectively. However, after 2015, as the number of poor reviews increased, the reply rates of poor and good reviews increased and decreased respectively. Across banks, poor reviews grew similarly and steadily over time with the main trajectory kicking off at 2018. The growth of good reviews is nosier but similar across challenger banks. Good reviews making up the minority for conventional banks is further shown when comparing the marginal distribution between good and poor reviews over time. 2010 corresponds to the earliest review year whereby conventional banks, like HSBC and NatWest, started to enter TrustPilot with challenger banks, like Wise, entering shortly after in 2012. Figure 26 considers review lengths when comparing banks.

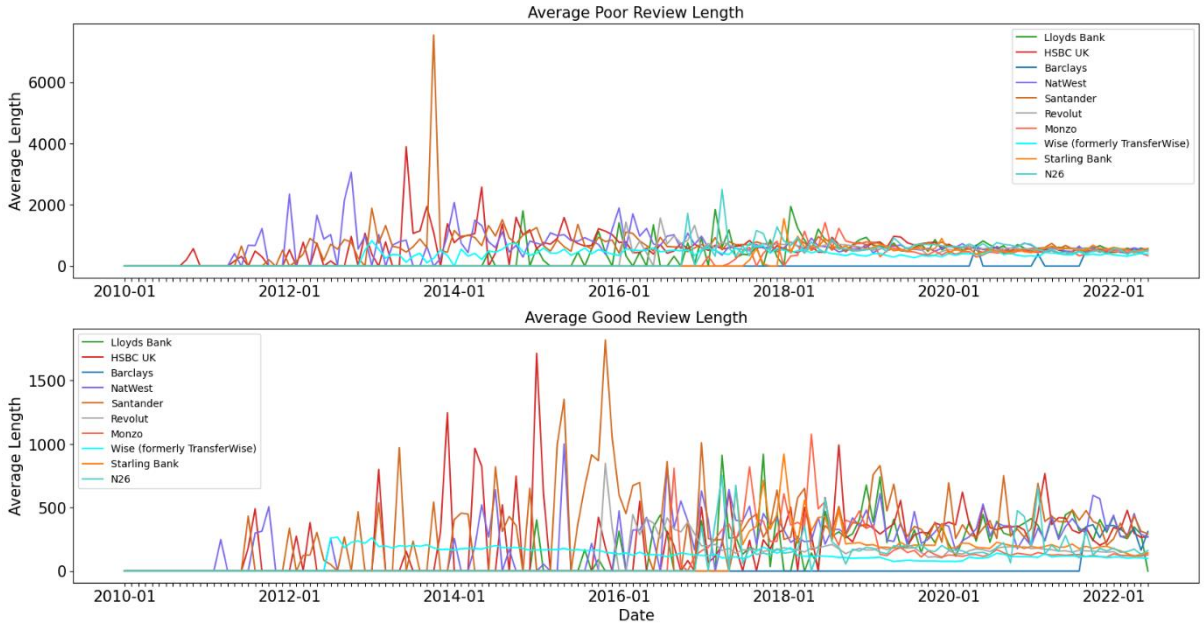


Figure 26: Average review length over time by rating

Generally, review lengths are comparable across bank types and respective rating. One noteworthy difference is the occurrence of more frequent peaks for conventional banks – for example poor review lengths of 7000 and 4000 nearing the end of 2013 for Santander and HSBC respectively. Reviews tend to be marginally longer earlier on over both ratings; observing before and after 2016 for example. Considering rating, poor and good reviews on average fluctuate around review lengths of 1000 and 500 respectively, indicating structural differences between ratings – potentially stemming from emotional urges to rant and elaborate about poor experiences rather than good ones.

4.3.2. User Demographic

This subsection assesses the user demographic between banks as a potential cause for the difference in distribution and number of reviews across bank types. Given the disparities between business models and product offerings, possible explanatory differences in the distribution of gender, age, and location were investigated. Figure 27 evaluates the plausibility of obtaining a distribution of user age by summarising the number users with and without profile photos.

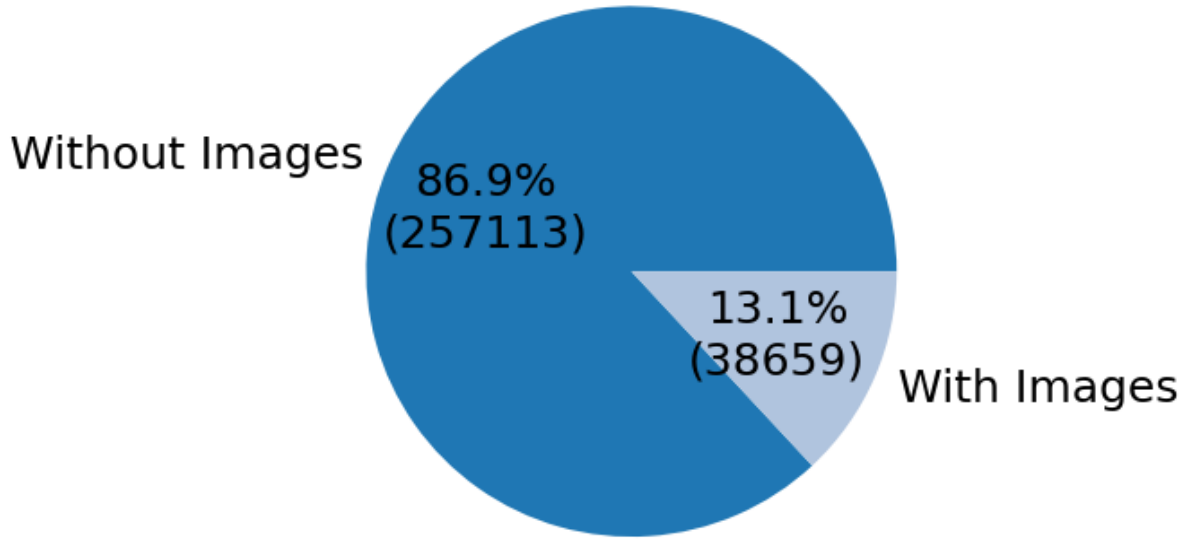
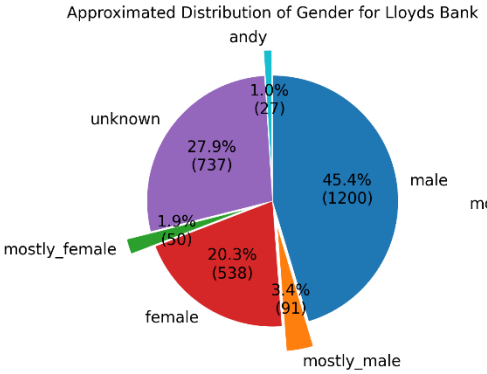
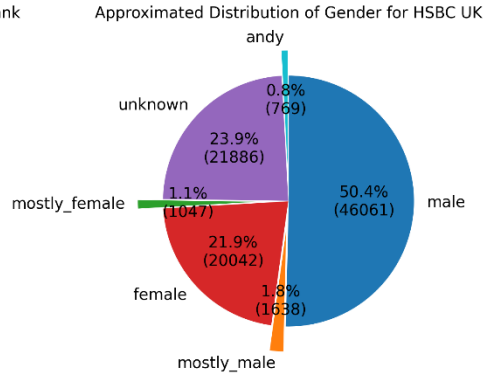


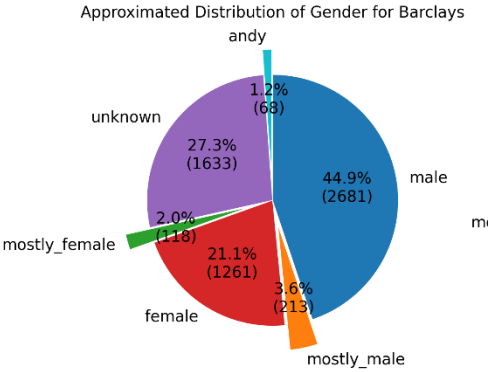
Figure 27: Distribution of users with and without profile photos

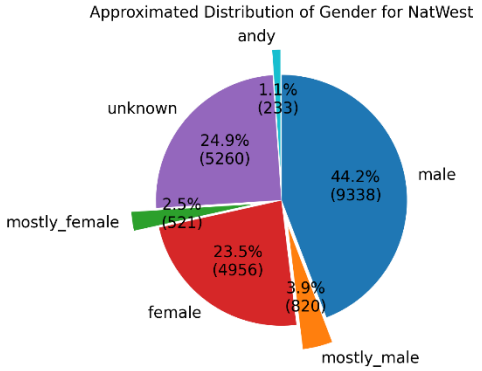
While usernames are mandatory on TrustPilot, profile photos are posted at a user's discretion. User profile photos were considered to supplement the approximation of a user's age, as mentioned in Section 3.3, but Figure 27 shows the difficulties with this strategy, due to only having over an eighth of profile photos. From the 13.1%, photos which are pixelated or do not include the user (e.g. a photo of a beach) also provided challenges for discerning a user's age and realistically results in a fraction of usable photos. The fraction of usable photos may not be reflective of the entire demographic and only reflect upon a portion of it, leading to false insights and conclusions. Figure 28 continues to study the user demographic through an approximated gender distribution using gender-guesser.

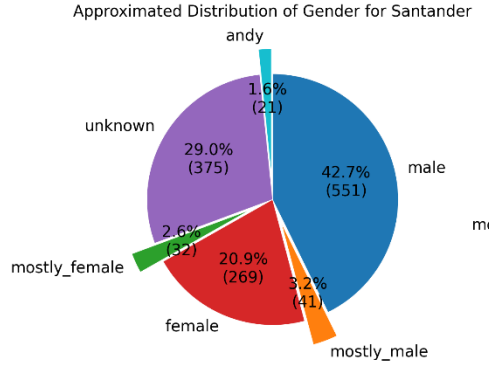
Gender-guesser assigns a gender to a user through their usernames and tags it as male, female, mostly male, mostly female, androgynous ('andy') or unknown if the gender cannot be identified. The gender distribution across banks are similar with males (and mostly males) making up between 45% and 53% of its respective users. Females (and mostly females) constitute 20% to 25% of each bank's users and an overlap of around 1% between genders (i.e. names which are common between both genders – androgynous names). The remaining users form 24% to 29% of the dataset whose gender was not identified. Though it could be a result of uncommon and/or foreign names, pseudonyms (i.e. users not using actual names for their usernames) could be a factor as well. The overall gender distribution between bank types are similar and marginal differences in distributions between banks may be due to distinctions in services or offerings (e.g. product, app or web interface). Whereas underlying biases potentially attribute towards differences in proportion of males and females within each bank - the proportion of male and females on TrustPilot as a whole, for example.

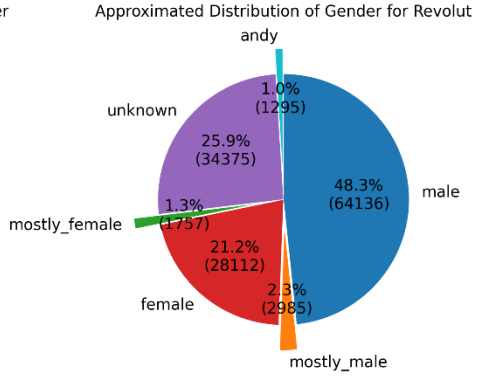

Approximated Distribution of Gender for Lloyds Bank
andy
1.0%
(27)
unknown
27.9%
(737)
45.4%
(1200)
male
1.9%
(50)
mostly_female
20.3%
(538)
3.4%
(91)
female
mostly_male

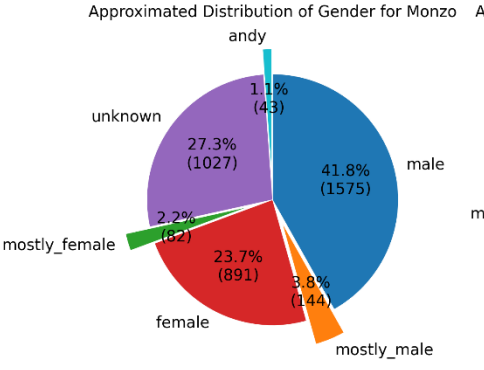

Approximated Distribution of Gender for HSBC UK
andy
unknown
0.8%
(769)
23.9%
(21886)
1.1%
(1047)
mostly_female
50.4%
(46061)
male
21.9%
(20042)
female
1.8%
(1638)
mostly_male

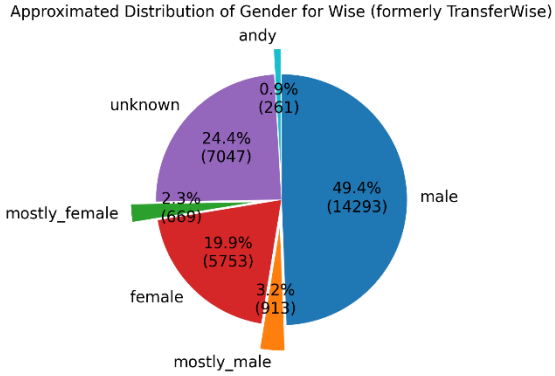

Approximated Distribution of Gender for Barclays
andy
1.2%
(68)
unknown
27.3%
(1633)
44.9%
(2681)
male
2.0%
(118)
mostly_female
21.1%
(1261)
3.6%
(213)
female
mostly_male

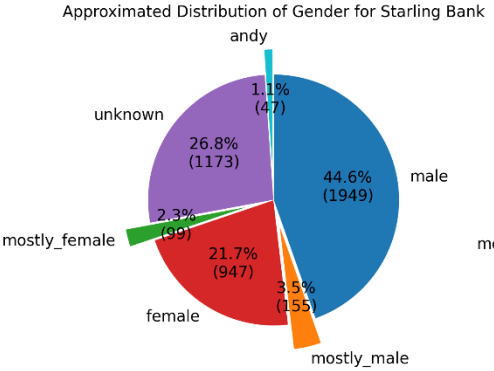

Approximated Distribution of Gender for NatWest
andy
1.1%
(233)
unknown
24.9%
(5260)
44.2%
(9338)
male
2.5%
(521)
mostly_female
23.5%
(4956)
3.9%
(820)
female
mostly_male


Approximated Distribution of Gender for Santander
andy
1.6%
(21)
unknown
29.0%
(375)
42.7%
(551)
male
2.6%
(32)
mostly_female
20.9%
(269)
3.2%
(41)
female
mostly_male


Approximated Distribution of Gender for Revolut
andy
1.0%
(1295)
unknown
25.9%
(34375)
48.3%
(64136)
male
1.3%
(1757)
mostly_female
21.2%
(28112)
2.3%
(2985)
female
mostly_male


Approximated Distribution of Gender for Monzo
andy
1.1%
(43)
unknown
27.3%
(1027)
41.8%
(1575)
male
2.2%
(82)
mostly_female
23.7%
(891)
3.8%
(144)
female
mostly_male


Approximated Distribution of Gender for Wise (formerly TransferWise)
andy
0.9%
(261)
unknown
24.4%
(7047)
49.4%
(14293)
male
2.3%
(669)
mostly_female
19.9%
(5753)
3.2%
(913)
female
mostly_male


Approximated Distribution of Gender for Starling Bank
andy
1.1%
(47)
unknown
26.8%
(1173)
44.6%
(1949)
male
2.3%
(99)
mostly_female
21.7%
(947)
3.5%
(155)
female
mostly_male

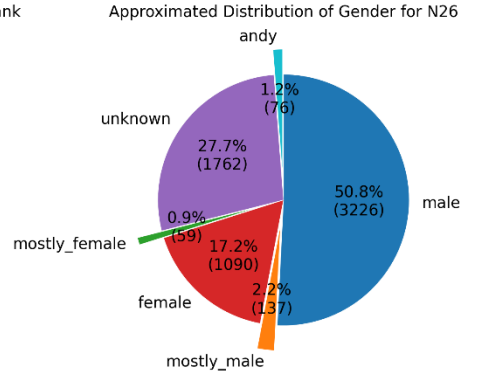

Approximated Distribution of Gender for N26
andy
1.2%
(76)
unknown
27.7%
(1762)
50.8%
(3226)
male
0.9%
(59)
mostly_female
17.2%
(1090)
2.2%
(137)
female
mostly_male

Figure 28: Approximated distribution of gender by bank

Figure 29 adds an additional dimension to user demographic by outlining the distribution of user locations for banks by deriving the 30 most common countries the reviews are from. From it, slight variations in distribution of user locations may be associated to operations, business model, strategy, and product offerings. Figure 29 shows the distributions for half of the banks (i.e. HSBC, Monzo, Revolut, Wise and N26), the remaining distributions of the remaining banks follow an identical distribution to HSBC and Monzo, as seen in Appendix 9.6. For example, fully-featured banks (i.e. conventional banks and Monzo) share close to equivalent user locations – Great Britain making up close to all user locations. However, N26 which is also a fully-featured bank, displayed a larger range of user locations – Germany, France, and Spain for instance. While N26 offers few additional products (e.g. on-demand insurance coverage), the country of operations may explain the dissimilarity – the banks under the former are UK-based (origination and operations) while N26 is German-based.

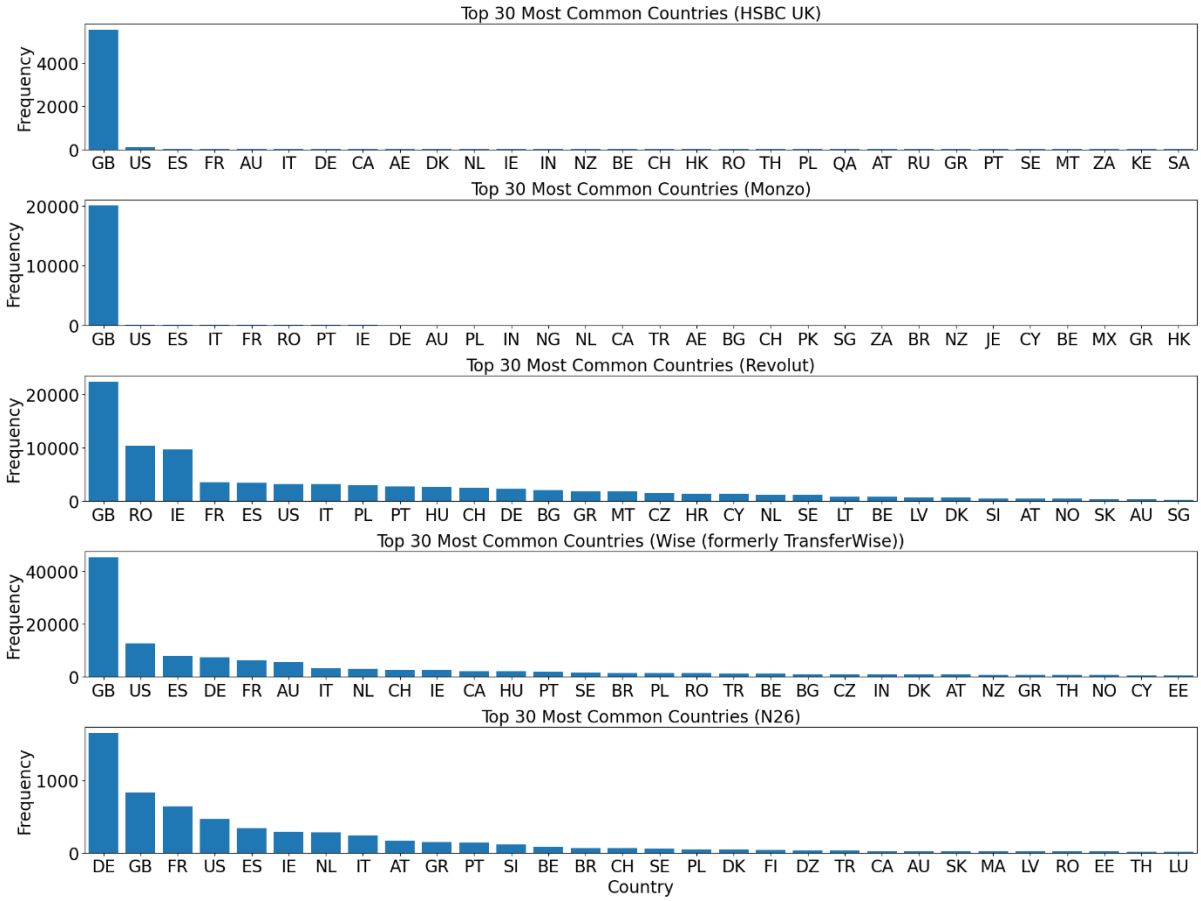


Figure 29: Distribution of user location by bank for HSBC, Monzo, Revolut, Wise and N26

Considering Revolut and Wise, both distributions deviate from the ones observed for fully-featured bank but are not identical either. The business models of Revolut and Wise classifies both as iterations of Fintech companies, whereby neither are fully-fledged banks but offer financial products. Wise's products surround foreign currencies (holding, receiving, sending and spending) resulting in significant

number of users outside of Europe – users from America, Canada, and Australia for instance (though users from Great Britain still constitute as the largest demographic). Revolut on the other hand, is made up predominantly of users from Great Britain, along with significant number of users from Romania and Ireland – around 10,000 users for both. It offers traditional banking products (e.g. bank accounts and debit cards) in addition to more distinct products like under-18 accounts. Its differing user demographic, compared to fully-featured banks and Wise, may have resulted from its business strategy - in 2015 and 2018, Revolut placed heavy efforts in entering the Irish and Romanian market respectively.

4.3.3. Unigram and Bigram Distribution

Evaluation of unigram and bigram distributions is a helpful method to finalise the list of stop words by including or removing new words with each iteration. Initially, the distribution of unigrams and bigrams will be littered with common words which provide little meaning but with each iteration, fewer stop words appear and a preliminary outlook on potential topics can be determined. Figure 30 charts the initial unigram and bigram distribution, prior to stop word removal, over the entire dataset (across good and poor reviews) by visualizing the top 20 most common unigrams and bigrams.

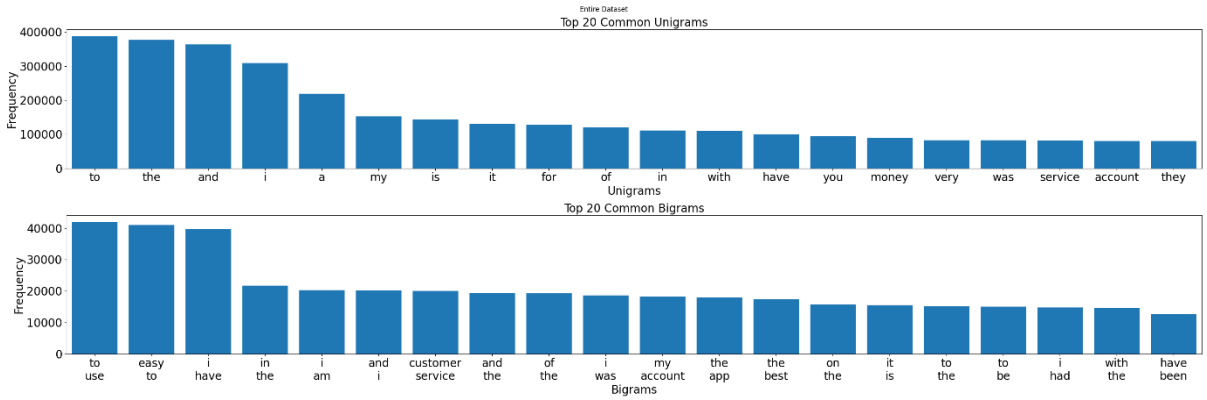


Figure 30: Unigram and bigram distribution over the entire dataset before stop word removal

The distributions contain many occurrences of commonly occurring words (and combination of words) like “to”, “the”, and “in the”. These words are tagged as stop words and have little implication on the meaning of a sentence or phrase and if excluded, alters the context and semantic insignificantly. Conversely, other common occurring words observed like “service”, “money”, “account”, and “easy” are not tagged as stop words. These words are either in reference to potential topics (e.g. customer service, money transfer, account opening) or sentiment (e.g. easy to use). In relation to the latter, words which convey the negation of a sentiment or satisfaction are also kept. Words like this include the word: “not”, whereby if excluded changes the semantic and sentiment of a sentence – for example, removing “not” from the negative sentiment phrase: “customer service team was not friendly”, alters it into a positive sentiment phrase: “customer service team was friendly”. Figure 31 shows similar unigram and bigram distributions as Figure 30, after the removal of stop words.

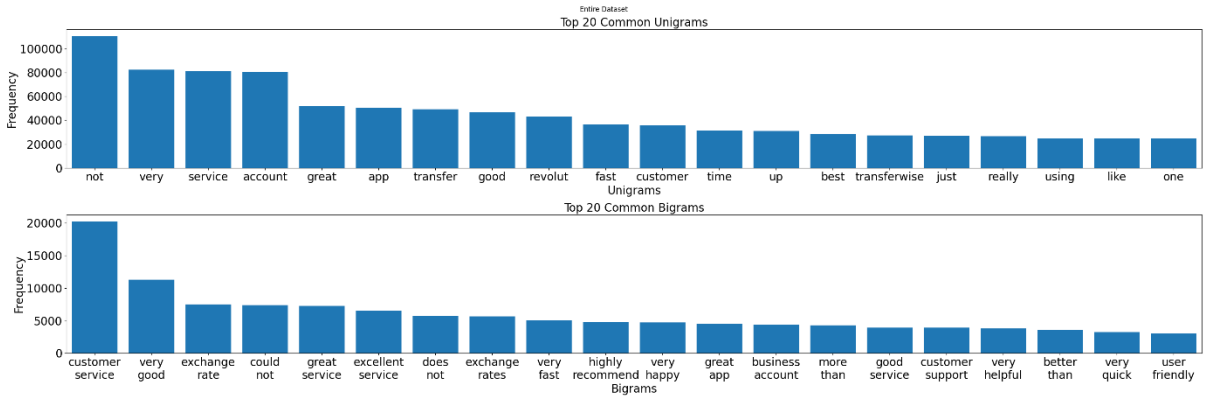
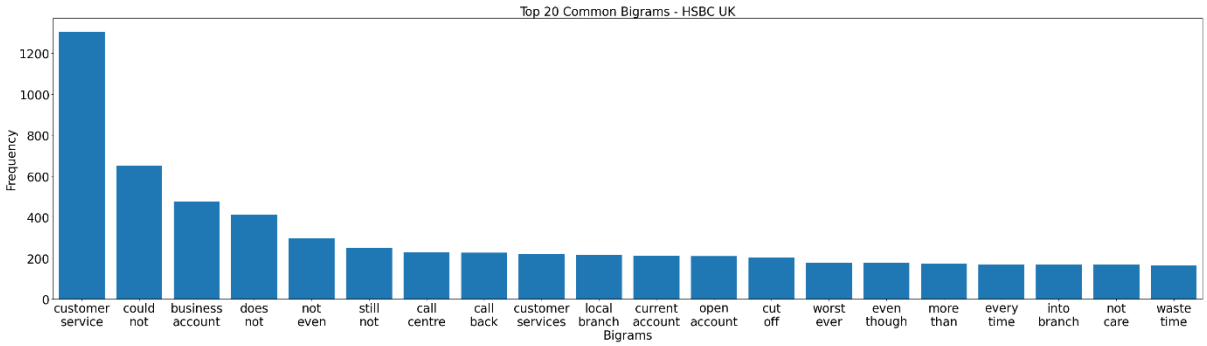


Figure 31: Unigram and bigram distribution over the entire dataset after stop word removal

After the removal of stop words, potential topics and sentiment are evident. With the former, customer service is most frequently cited along with other elements that a bank would monitor – exchange rates, business accounts, and user experience on online platforms. The latter shows positive sentiment with user satisfaction through recurrent uses of: “great”, “fast”, “very helpful”, and “good service”. Given that the dataset is predominantly made up of good reviews, this is expected. However, the repeated use of “not” may be an indicator stemming from poor reviews – “not happy” and “not pleased” for instance. Bank names are also observed with Revolut and Wise being persistently mentioned. Though it is unclear if these arise from their own respective reviews (e.g. Revolut reviews mentioning Revolut), the numerous uses of “better than” may signify comparisons between banks. To account for the former, bank names are removed from its own respective reviews, as mentioned in Section 3.1. The full list of stop words is listed in Appendix 9.4.

The above plots show the importance of unigram and bigram distributions separately. While unigram distributions aid the stop word evaluation process and allude to potential topics, these topics are more visible in bigram distributions. To further identify and compare preliminary topics between banks, Figure 32 charts the bigram distribution over poor reviews using HSBC and Wise as representatives to each bank type. The bigram distributions for each bank are displayed in Appendix 9.7.



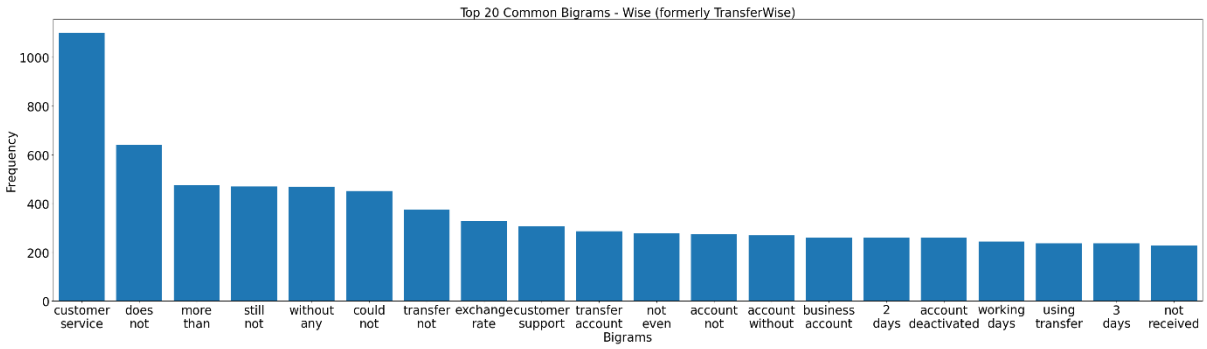


Figure 32: Bigram distribution of poor reviews for HSBC and Wise after stop word removal

For both banks (and over each bank), users are most dissatisfied with their customer service and negative sentiments shadow each bigram distribution. With bigram distributions computed over poor reviews and customer service being an overarching topic which umbrellas other components, it is unsurprising. Components like poor turnover times and unresolved issues are referenced to and hinted at when observing the frequent use of: "3 days" and "call back", respectively. Topics and context specific to each bank are also identified – for example: "current account" and "local branch" for HSBC, "exchange rate" and "transfer account" for Wise. These preliminary topics provide good foundational results prior to implementing the methods described in Section 3.4, whilst also highlighting the importance of manual judgement.

Despite the differences in review distribution, the user demographic and preliminary topics are similar; supporting the plausibility of extending trained models over the entire dataset, rather than segmenting into bank types. Though, given the variation in review distribution and bank specific topics, considerations to the comprehensiveness of the training dataset will need to be made to ensure a majority of topics are represented – cross currency transfers specific to Wise for instance. Without it, classification of underrepresented topics may be inaccurate or not identified at all.

4.4. Feature Extraction

Smaller experiments using review sentences from Monzo were first performed as a proof-of-concept prior to extrapolating and scaling over a larger subsample. Experiments with word embedding models and varying text input formats were conducted in parallel. Table 6 presents the results of the former.

Word Embedding Model	Model Base	CPU Time (s)	Wall Time (s)
Universal-sentence-encoder (TensorFlow Hub, no date a)	Deep Averaging Network (DAN)	3.23	1.99
Universal-sentence-encoder-large (TensorFlow Hub, no date b)	Transformer	339	83

Table 6: Word embedding model experiments

With the transformer-based model, processing time took over 100 times the amount of time when compared to the DAN-based model. Differences in operational complexity are the causes of the difference - $O(n^2)$ and $O(n)$ for the transformer and DAN-based approach respectively - but despite this, the transformer-based model produces better transfer task performance (Cer *et al.*, 2018). Processing times also exceeded wall times due to multithreading/parallelisation. Despite the difference in magnitude, the processing time of the transformer-based model fell within an acceptable threshold (between 40 to 60 clusters) and was selected as the word embedding model of choice (though considerations of scalability will need to be considered for future works), yielding a 512-dimensional vector representation of the review sentences.

The 27 possible hyperparameter combinations were evaluated by performing visual analysis in combination with intuitive consideration of their relative validity. The optimal hyperparameter combination would be one which visually represents well-segmented clusters whilst balancing it against the relative validity. Figure 33 and Table 7 lays out visual plots of three hyperparameter combinations, their hyperparameter values, relative validities, number of clusters.

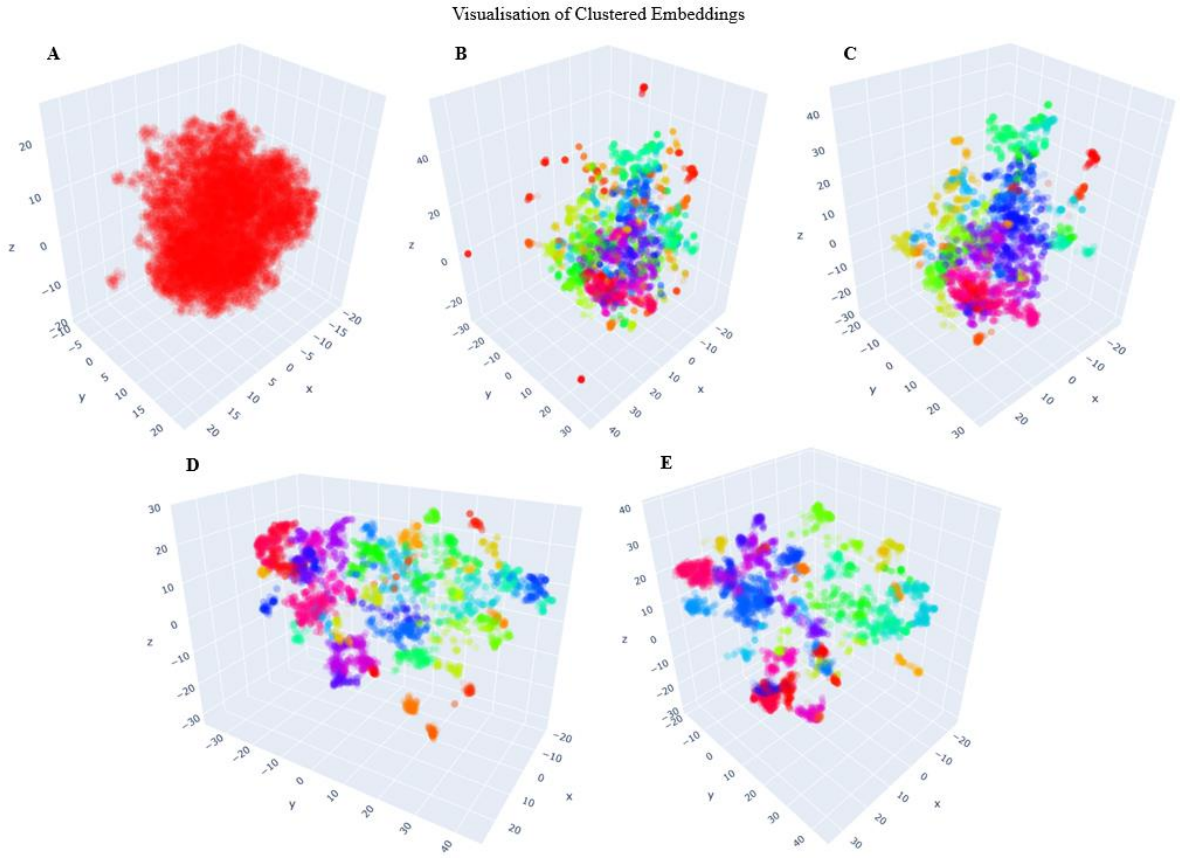


Figure 33: 3-dimensional plots of clustered embeddings over different hyperparameter combinations: A, B, and C

Hyperparameter Combination	Hyperparameter Values		Metrics
	UMAP	HDBSCAN	

	n_neighbors	min_dist	min_cluster	Relative Validity	Number of Clusters
A	15	0.99	15	0.576	2
B	15	0.0	15	0.292	304
C	15	0.0	60	0.161	60
D	50	0.0	60	0.211	66
E	100	0.0	60	0.189	53

Table 7: Hyperparameter combination values and results - Monzo

Basing decisions on relative validity alone points toward hyperparameter combination A as the best amongst the four. However, upon visual inspection and determining the number of clusters, the high relative validity score is a result of the lack of segmentation and clustering amongst the embeddings – only forming 2 clusters. The relative validity is an approximation to the DBCV score which computes densities within and between clusters (Moulavi *et al.,* 2014). With only 2 large clusters, the density within clusters may be the driving factor for the high relative validity. For this problem scenario, the quality and number of clusters were preferred as the main determining factor (to ensure representation of various topics) while relative validity scores provide a secondary measure.

Reducing the min_dist to 0 (hyperparameter combination B), points are now loosely packed resulting in more segmented and clumpier clusters at the expense of its relative validity score. Though the low value of min_dist prevents the preservation of global topological structure over the embeddings, finer topological structure is preferred to generate cluster representation of topics - different clusters are shown through the variation in colour. However, 304 clusters may be product of high levels of granularity (with large amounts of overlap in topics between clusters and/or several non-sensical clusters) and lend itself to challenges (e.g. time constraints) during cluster and topic evaluation.

Increasing the min_cluster to 60 (combination C), more points are now required to constitute towards a cluster. Consequentially, smaller clusters are merged to form larger clusters, amounting to fewer number of clusters overall. Despite the lowered relative validity scores, the number of clusters strike a balance between having too few clusters (combination A) and too many clusters (combination B). Further combinations are examined to identify better results.

Bumping the n_neighbors to 50 (combination D), a broader view of the data is considered when performing UMAP. Global structures are now preferred over local structures and the loss of local structure is represented by the separation of closely related colours. For example, in B and C, clusters with blue hues tend to be closer together but in D, these clusters are separated with some being closer towards clusters with red hues instead. As a by-product, the number of clusters remain at a manageable amount whilst still maintaining a higher relative validity score.

Raising the n_neighbors further to 100 (combination E), the number of clusters stay at a reasonable amount but with lower relative validity scores. However, upon visual inspection the embeddings are segmented to more distinct clusters, loosely distributed spatially.

Applying topic modelling techniques onto the clusters (to get a set of topics for each cluster), the quality of clustering over hyperparameter combinations are evaluated further. Figure 34 plots the distribution of clusters and topics for both combinations, D and E. Both LDA and NMF were used to generate topics over each cluster but for comparison, Figure 34 only plots the distribution of the top 15 largest clusters and the top 3 corresponding topics over each cluster produced by NMF.

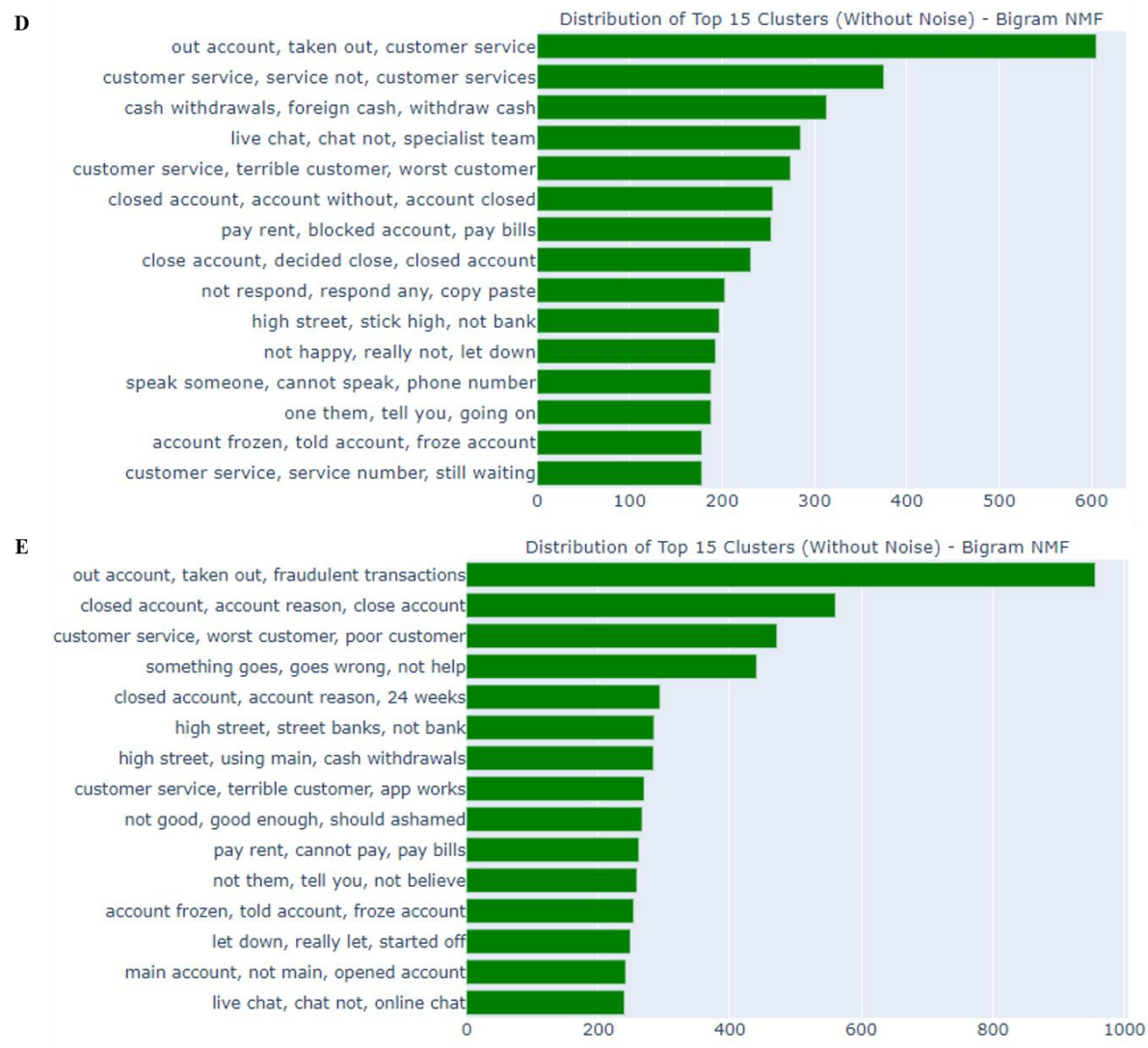


Figure 34: Topic and cluster distribution over hyperparameter combinations: D and E

Both D and E showcase similar niche topics like frozen accounts, account closures, payment issues and broader topics like customer service. E, however, identifies additional topics surrounding fraud and business accounts - topics which are not found in D (though maybe in a smaller cluster beyond the top 15 largest clusters). The larger cluster sizes in E also further suggest the aggregation of smaller and

similar clusters to larger singular ones; reducing the amount of manual reallocation of clusters. For these reasons, Combination E was selected as the optimal combination. The hyperparameter combinations and results of all 27 combinations can be found in Appendix 9.8. Additional experiments on different types of text input were conducted prior to applying the techniques onto a larger subsample and are shown in Figure 35 and Table 8.

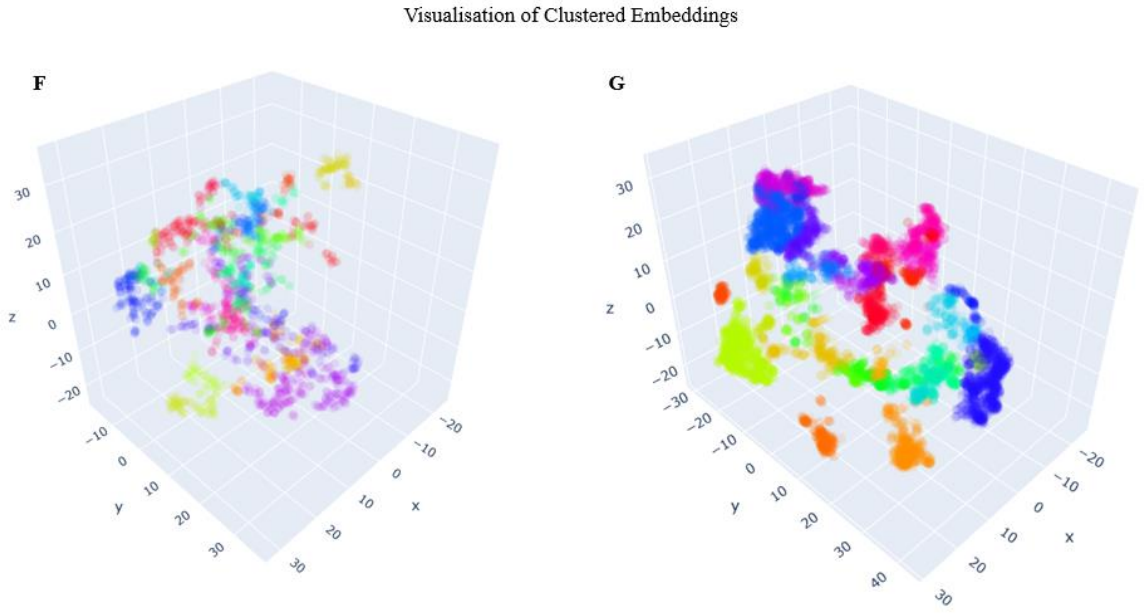


Figure 35: 3-dimensional plots of clustered embeddings over different text inputs: D and E

Experiment	Text Input	Metrics	
		Relative Validity	Number of Clusters
F	Entire reviews	0.166	55
G	Review sentences with minimal processing	0.258	43

Table 8: Text input experiment results - Monzo

Experiments F and G employ the same hyperparameter combination as E but explore with different text inputs. Experiments F and G takes as input the entire cleaned reviews (without sentence tokenisation) and review sentences with minimal processing (without data cleaning steps like stop word removal) respectively. By using the same hyperparameter combination as E, the implications of hyperparameters are controlled for and variation in quality of dimensionality reduction and clustering are isolated to the difference in inputs.

In experiment F, the total number of clusters stayed relatively similar with 2 additional clusters. However, from visual comparisons, segmentation of clusters is less distinct likely due to the dilution of word embeddings from considering a collection of sentences (review) instead of individual sentences (tokenised sentences). For example, multiple topics or contradictory sentiments (e.g. a positive

experience caveated by a negative one) may amount to convoluted reviews, semantic, and inaccurate embeddings. Relying upon entire reviews rather than review sentences also reduces the size of the dataset shown through the lack of colour saturation within the 3-dimensional plot. Smaller dataset, in addition to diluted embeddings, results in fewer training points amounting to potentially lower performance and non-robust models. In total, the dataset would be reduced from 233,699 review sentences to 43,365 reviews – nearly a fifth of the size.

Experiment G showed better results than E and F through the quality of clustering, accompanied by the higher relative validity score. The number of clusters continue to sit at a manageable threshold while the embeddings are more obviously segmented with clear and distinct clusters. Figure 36 shows the distribution of clusters and topics. In addition to the topics identified by E, topics like overdraft and credit score are seen. More specific topics which fall under large overarching ones are also identified clearly. For example, differentiated topics like long turnover times, poor live chat, and in-app experience are identified but fall within the realm of customer experience and service. Thus, when scaling the methods highlighted in Section 3.4, review sentences with minimal processing should be parsed as inputs rather than processed review sentences – reiterating the findings from Cer *et al.* (2018).

Figure 36: Topic and cluster distribution over hyperparameter combinations G

Review sentences from Revolut were chosen as the larger subsample due to the number of review sentences and extensiveness in services and offerings – ensuring that a broad range of topics are represented. Extending the methods above, a 3-dimensional representation of clustered embeddings are obtained and illustrated in Figure 37. Following a consistent set of arguments, hyperparameter tuning was performed similarly but over larger values of min_cluster (due to the increased dataset size) – 100, 200 and 300 specifically.

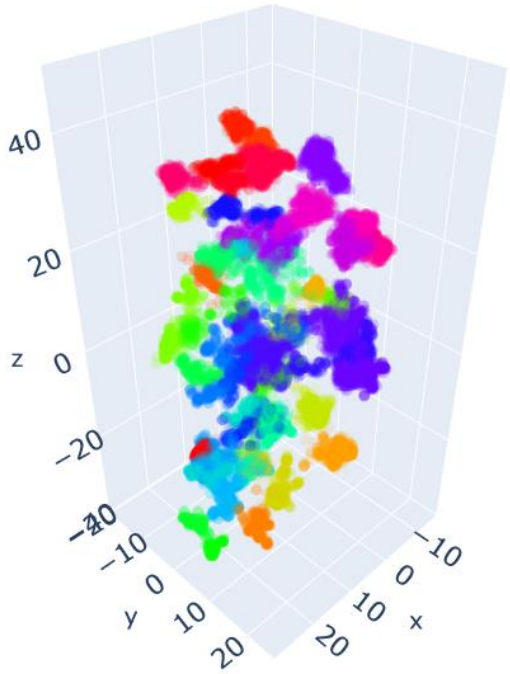


Figure 37: 3-dimensional plots of clustered embeddings – Revolut

Table 9 presents the optimal hyperparameter chosen, its results, and the results of all 27 hyperparameter combinations are listed in Appendix 9.9.

Hyperparameter Values			Metrics	
UMAP		HDBSCAN	Relative Validity	Number of Clusters
n_neighbors	min_dist	min_cluster		
100	0.0	300	0.209	49

Table 9: Hyperparameter combination values and results – Revolut

Through LDA and NMF, a series of topics are tagged onto each cluster, allowing for a main topic to be manually determined and assigned. Examples over a subset of the clusters are shown in Table 10 while the full listing of clusters are presented in Appendix 9.10.

Cluster Number	Cluster Size	NMF Topics	LDA Topics	Assigned Topic
1	647	crypto currency, amount crypto, buy crypto, not buy, does not, find out	cryptocurrency exchange, premium account, buy crypto, cannot send, crypto currency, cannot transfer	Crypto
2	302	trading platform, full year, buy sell, cannot buy, stock trading, gme amc	buy stocks, very very, trading account, asset value, trading platform, buy sell	Stock
5	900	using app, started using, app ever, worst app, app good, service app	about app, app using, app ever, good app,	App interface

			downloaded app, app not	
10	320	account blocked, blocked since, blocked account, account took, still blocked, account still	blocked account, why blocked, just blocked, got blocked, account blocked, block account	Blocked account
44	545	exchange rates, good exchange, exchange rate, best exchange, currency exchange, like currency	currency exchange, foreign currency, multiple currencies, more than, exchange rates, exchange rate	Exchange rates
47	463	euro account, account euro, uk account, transfer uk, gbp account, even though	euro account, uk account, gbp account, into account, eur gbp, top up	Cross currency transfers

Table 10: Example of cluster topics

Specific and niche topics are identified through the concurrence of generated NMF and LDA topics. For example, the topics produced by NMF and LDA in cluster 1 provide clear indication of negative experiences surrounding cryptocurrencies – both, purchasing and transfers. Unaccounted topics like stock are identified (reiterating the advantages of a non-parametric approach), whilst topics which are more prevalent in opposing banks are still identified through Revolut's reviews – cross currency transfers for instance. Differences in cluster size also provide prioritisation opportunities from the perspective of banks – tending to app interface may lead to larger wins compared to issues surrounding buying and selling of stock. Manual assignation of topics contributes towards evaluating the regrouping of clusters.

Table 11 exhibits a subset of clusters regrouped as noise.

Cluster Number	Cluster Size	NMF Topics	LDA Topics	Assigned Topic	Regrouped Topic
16	316	bad experience, very bad, worst experience, experience ever, terrible experience, most terrible	awful experience, share experience, bad experience, experience not, experience ever, worst experience	Dissatisfaction	Noise
17	416	very disappointed, very very, not happy, really not, very frustrating, frustrating experience	very disappointed, really disappointed, not happy, extremely frustrating, very frustrating, very happy	Dissatisfaction	
22	304	goes wrong, something goes, great until, until problem, idea how, absolutely idea	went wrong, big mistake, goes wrong, something goes, not same, same mistake	Undetermined	

24	1045	not work, does not, top up, wanted top, never again, hard earned	again again, not rely, does not, not work, top up, first time	Undetermined	

Table 11: Example of cluster topics regrouped as noise

Non-sensical clusters are categorised into two categories – clusters which provide no indication on potential areas of improvements and clusters whose topics are too vague and indistinguishable, leading to undetermined topics. Clusters 16 and 17 falls into the former whereby its review sentences circle around dissatisfaction through bad experiences and disappointment respectively. Both lack elaboration on the service or offering which caused dissatisfaction and only supplies the same information inferred from the rating of reviews. From the perspective of banks, these clusters prevent executable actions for improvement. On the other hand, cluster 22 and 24 fall into the latter of non-sensical clusters. Consistent and apparent topics are absent, and topics appear noisy. A coherent topic was not assigned to these clusters and were instead tagged as "undetermined" prior to regrouping it under noise. Contrastingly,

Table 12 presents a subset of clusters with similar topics.

Cluster Number	Cluster Size	NMF Topics	LDA Topics	Assigned Topic	Regrouped Topic
6	461	live agent, touch live, looking agent, still looking, different agents, app chat	live agent, agent not, different agents, live agent, live agent, looking agent	Live chat experience	Live chat experience
30	491	live chat, chat not, via chat, chat team, online chat, every time	live chat, chat not, contacted chat, chat history, online chat, every time	Live chat experience	
25	482	sent documents, documents requested, more documents, asking more, send documents, not find	every time, submitted documents, requested documents, same documents, send documents, more documents	Documents request	Tedious paperwork (request of documents)
40	334	tax return, asked tax, tax returns, company tax, asked provide, invoices etc	months statements, last three, tax return, tax returns, asked provide, last months	Invoice request	

Table 12: Example of similar cluster topics

Similarities between clusters and topics are classified into two categories – clusters which are explicitly similar (i.e. same topic) and clusters which fall under a similar overarching topic. Clusters 6 and 30 are an example of the former. From the NMF and LDA topics, both clusters are made up of review sentences surrounding live chat experiences and is further supported when evaluating sentences from each cluster with the highest clustering probabilities, as shown in Table 13. With both, challenges

surrounding chatting with live agents are highlighted – long response times, incompetence, and poorly designed user experience.

Sentence Number	Pre-processed Sentence		Soft Clustering Probability	
	Cluster 6	Cluster 30	Cluster 6	Cluster 30
1	I spent two days with your agents, trying to have some feedback.	They have not even read the detailed chat history shared with them on my prolonged issued.	0.766	0.668
2	Got a message on app chat, was asked a couple of security questions, waited over an hour, got reply that agent was finishing shift and somebody else would take over.	Each time contacting Revolut I have to explain the case anew to a live chat agent despite having a formal complaint reference number.	0.755	0.508
3	Your agent was nice but could not provide information about my card.	However unlike email you have to sit and wait for the reply, even if it takes hours, tapping the screen to keep the chat alive, otherwise when someone comes across your question you'll get a response "I can see you're not available to chat now" and no answer.	0.720	0.469
4	They just follow all the protocols given to them and repeat the same sentences over and over again without actually checking up exactly what’s going on.	So, here I am on chat waiting for a response and just kept hanging such terrible service.	0.644	0.456
5	I contacted live chat and surprisingly got a quick response but the agent kept	I have just received a message and tapped it straight away to reply but	0.524	0.400

	disappearing and I was left hanging.	every time I do so the chat vanishes, then I eventually get the message "we haven't heard from you in a while so need to archive this chat".		

Table 13: Top 5 sentences with the highest soft clustering probability for clusters 6 and 30

Soft clustering probabilities also serve as indicators of confidence and sentences associated to higher clustering probabilities may provide clearer indication of topics over lower counterparts. Sentences with lower clustering probabilities are useful and act as a component to assessing clustering quality as well. These sentences may not be indicative of the topic or highlight an entirely different topic. However, evaluating the low cluster probability sentences yielded similar conclusions, as shown in Table 14. Review sentences continued to surround negative experiences with live agents (inaccessibility and incompetency) and low clustering probabilities may be a result of the interchangeability between “chat”, “agent”, and “bot” – words that are understood to be similar in the context of live chats through human judgement.

Sentence Number	**Pre-processed Sentence**		**Soft Clustering Probability**	
	Cluster 6	**Cluster 30**	**Cluster 6**	**Cluster 30**
1	Live agent at the help desk doesn't have this information as well, so I have no idea what did I pay for as premium member, shall I downgrade, ask for refund?!	You can’t call them directly, only “live chat” where I waiting for response already 3 days.	0.006	0.007
2	When you reach to speak to a live agent (not the bot) they are sometimes useless, you have to explain your problem 3 or 4 times before they understand a resolve it.	No one available on live chat and no facility to talk to anyone in person	0.006	0.010
3	My normal bank in the US gives me a phone	The "live chat" is a joke.	0.006	0.012

	representative almost immediately.			
4	Asked to speak to a manager, none available.	They are instructed to give a default reply that they have to right to keep them for 6 years.	0.007	0.013
5	Im stuck talkin to bots and I keep asking for live agent or someone to ring me but no one gets back to me they did this last time and I told them Im slitting my wrist and they gave me my money and police was sent to my house to see if Im good once I get my money back.	The issues I have had have only been very trivial and made a lot worse by a chat bot closing the issue instead of answering it or connecting with a live agent	0.007	0.014

Table 14: Top 5 sentences with the lowest soft clustering probability for clusters 6 and 30

Clusters 25 and 40 from Table 12 both represent distinct topics but fall under a similar overarching topic. While cluster 25 talks about transactional the request of documents for services (e.g. account opening), cluster 29 covers the request of invoices for taxes. However, from similar inspections as above, the review sentences from both surrounded similar reasoning – the continuous and frequent request of documents/invoices. Thus, both clusters were grouped together and retagged as additional fees. Through iterative exercises following a consistent series of arguments, clusters were grouped and later reassigned topics. The regrouped clusters prior to model development are laid out in Appendix 9.11.

4.5. Text Classification

Prior to training a series of binary random forest models, the distribution of cluster sizes is determined to ensure a robust dataset for training, validation and testing. Skewed distributions amount to potential underrepresentation (or overrepresentation) of certain clusters. Even within the negative class, a skewed distribution may result in non-robustness of models through biased predictions and lack of generalisability over an unseen dataset. The distribution of cluster sizes is plotted in Figure 38.

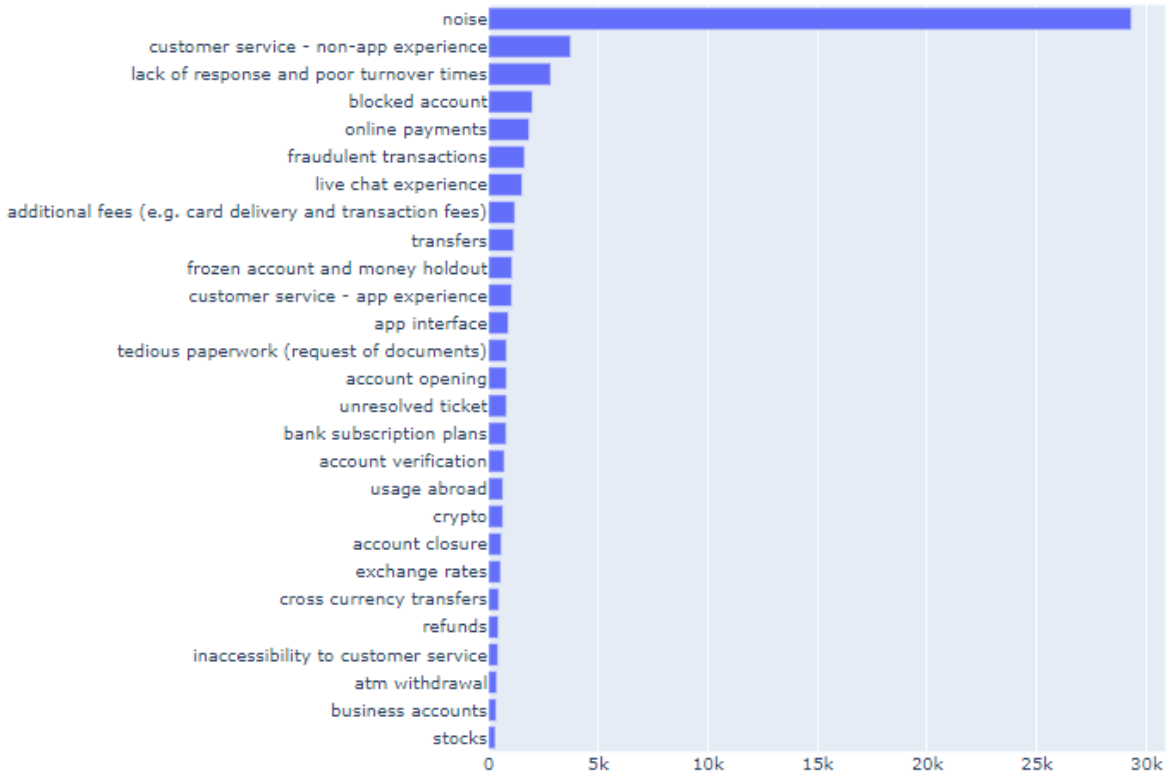


Figure 38: Ordered distribution of cluster sizes

The distribution of cluster sizes is skewed towards the noise cluster, with it making up majority of the dataset – 29,314 and 27,632 review sentences are tagged as noise and non-noise respectively. To allow for a more uniform distribution of clusters, the noise cluster is down-sampled (i.e. underrepresenting an overrepresented cluster) using random sampling to the average cluster size when excluding noise (1063). Down-sampling of review sentences within the noise cluster mitigates potential sampling biases of the negative class (e.g. sentences not under cluster 1 when fitting a binary random forest for cluster 1). Down-sampling was performed after splitting the dataset into training (80%) and testing (20%). The resulting distribution of clusters over the training set is presented in Figure 39. Noise is now less represented while customer service – non-app experience is the largest cluster.

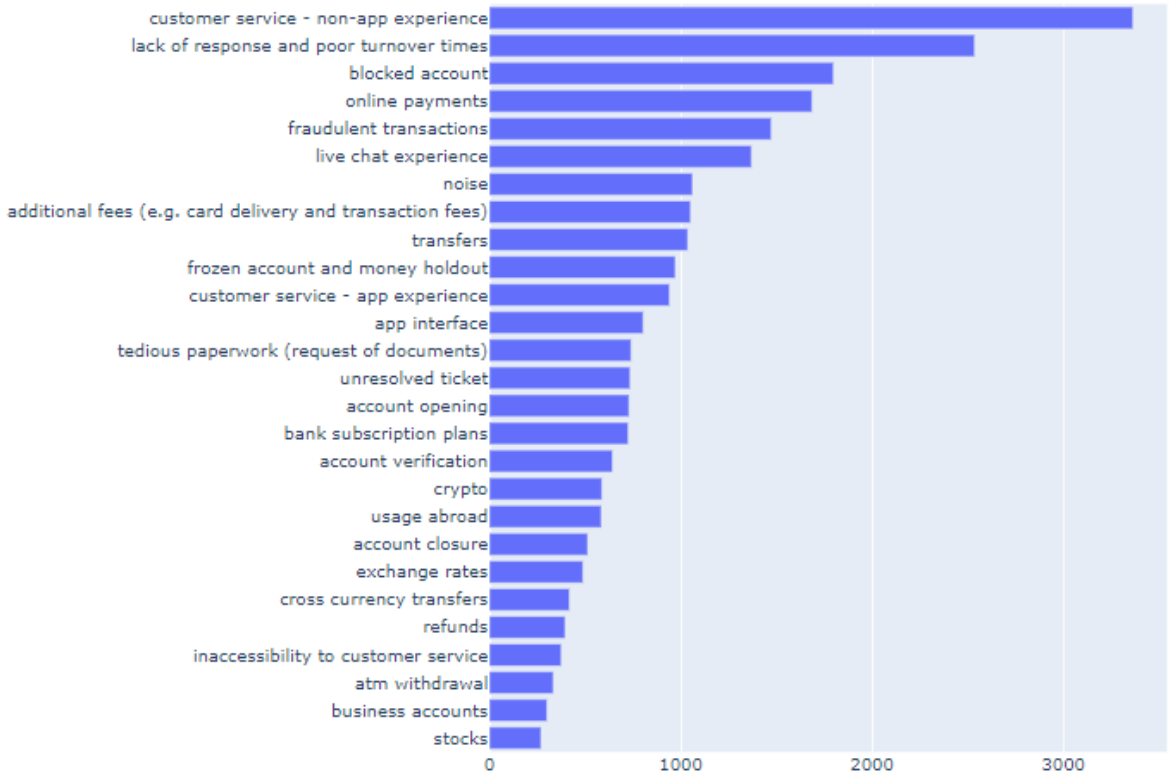


Figure 39: Ordered distribution of cluster sizes over training set

4.5.1. Binary Classification

A binary random forest classification model is trained over each cluster. With each, review sentences tagged as the given cluster are treated as the positive class and labelled as 1. Conversely, sentences not tagged as the given cluster are treated as the negative class and labelled as 0. For example, with cluster 1, sentences within cluster 1 are labelled as +1 while sentences within other clusters are labelled as 0. Developing binary classification models over the entire training set leads to biases in predictions due to differences in size between the positive and negative class. For instance, a binary classification model trained over a dataset comprised of 90% and 10% of negative and positive class respectively still achieves 90% accuracy when only predicting the negative class. Thus, the negative class was down-sampled (through random sampling) to the size of the positive class – forming an equally distributed dataset that is split into training and testing using an 80-20 split.

Experimenting over a single cluster, cluster 16 (customer service non-app experience), the 'n_estimators', 'max_depth' and 'max_features' were tuned using grid search; selecting the hyperparameter combination with the highest training accuracy. Subsequently, the random forest is retrained over 10 folds, predictions are aggregated and model parameters are selected from the fold with the highest validation accuracy. Table 15 and Table 16 displays the hyperparameter combination and model performance over training and validation respectively.

Cluster Number	Hyperparameter Values		
	n_estimators	max_depth	max_features
16	1000	100	25

Table 15: Optimal hyperparameter combination - random forest

Cluster Number	Dataset	Label	Metrics			
			Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
16	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	92	94	93
		1	93	94	93	94

Table 16: Binary random forest model performance

The optimal random forest model is one made up of 1000 decision trees which considers 25 features (of the 512 dimensions) at each split, with a maximum distance of 100 between root and leaf node. Correspondingly, the labels of all samples within the training set were correctly predicted resulting in 100% accuracy, precision, recall, and f1-score. Though it may be an indicator of overfitting, performance over the validation set only fell marginally short with an average of 93% across all metrics over both labels, 0 and 1. The marginally lower recall relative to precision over the validation set for the positive class suggests higher rates of misclassification within the positive class (false negatives) as compared to the negative class (false positives). Confusion matrices provide a clearer narrative and the confusion matrix over the validation set is presented in Figure 40.

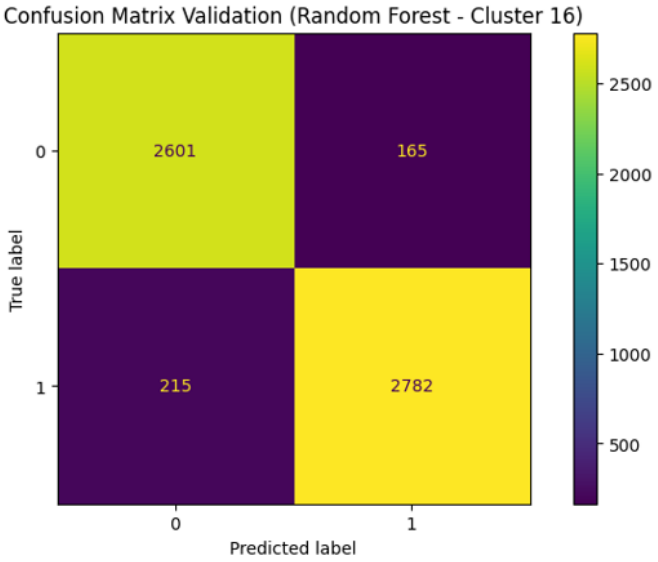


Figure 40: Confusion matrix over validation set

The number of false negatives exceed the number of false positives but the rate of false positives (relative to the total number of positives) is lower compared to the rate of false negatives. In this

instance, false negatives represent the sentences within cluster 16 which were tagged under a different cluster and false positives represent the sentences outside of cluster 16 which were tagged under cluster 16. Both result in false representation of clusters and poorly trained models further down the pipeline; potentially impacting decision making. For illustration, high rates of false positives may falsely indicate that user's experiences with customer service are good. The classification probabilities provide an additional method to evaluating false positives and negatives. Figure 41 plots the classification probability distribution over the training and testing set.

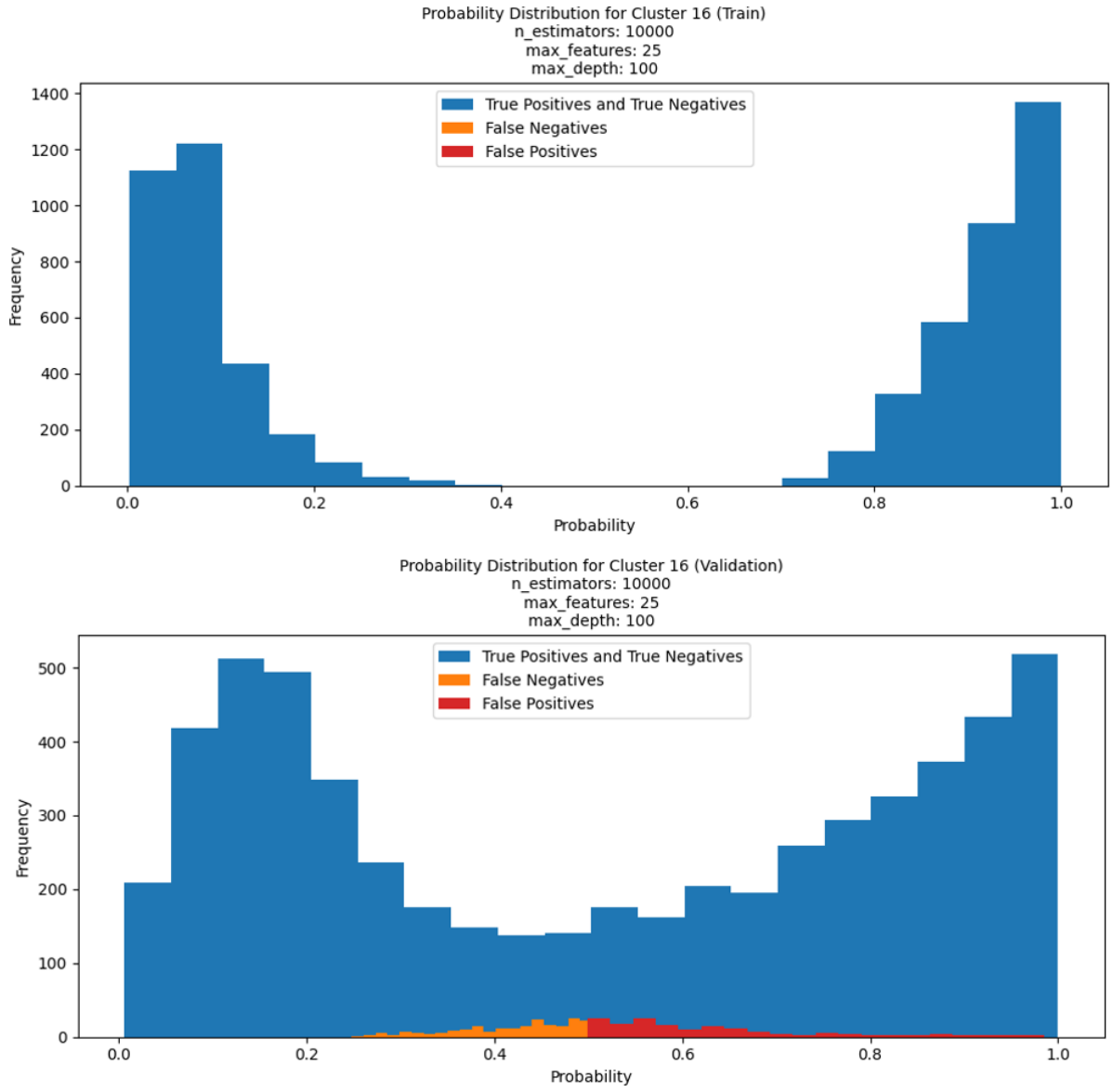


Figure 41: Classification probability distribution

The classification probabilities depict the proportion of votes amongst the decision trees within the random forest. For example, a classification probability of 0.80 suggests that 80% of decision trees voted for the positive class while the remaining 20% voted for the negative class. Comparing it over both datasets, the predictions over the training set are more decisive. The collection of decision trees

unanimously voted for or against the positive class, shown by the dual-peaks surrounding probabilities of 0 and 1. The distribution over the validation set also follows a multi-modal distribution (showing promise for the clustering and binary classification) but with more fringe decisions surrounding 0.4 to 0.6 – close to even split in votes. The majority of false positives and negatives form part of the fringe decisions. While high rates of false positives and false negatives of probabilities close to 1 and 0 respectively imply poor clustering, fringe decisions signify potential improvements through further regrouping of clusters. Hence, the false positives and negatives are further investigated through manual evaluation of the sentences with the highest and lowest classification probabilities respectively. The results are demonstrated in Table 17 and Table 18 respectively.

Sentence Number	Sentence	Original Cluster	Classification Probability
1	Customer Service is like a slow death.... lack of communication and never a straight answer.	14 (customer service - app experience)	0.985
2	And it is a shame that the customer service is so bad, no human person to talk to, and the chat from the app is totally unhelpful, nobody responding.	14 (customer service - app experience)	0.979
3	However, customer support is their big Achilles heel and without dedicated phone lines where you can talk to someone, all customer support must be performed through a woeful chat service.	14 (customer service - app experience)	0.969
4	But when asking for your help over your so-called “great customer service”, I have been told that a replacement of a new card has to be covered from my own side which is ridiculous.	14 (customer service - app experience)	0.950
5	I was met with a condescending tone from an impolite customer support agent.	14 (customer service - app experience)	0.943

Table 17: Top 5 false positives with the highest classification probability

Sentence Number	Sentence	Classification Probability
1	On the same day, I had another issue, as my bank account was linked to my Revolut app, and I tried to contact an agent for this new issue.	0.248
2	He also closed the chat on my face his name (Konrad).	0.266

3	I spoke to Jojo, Michelle and Ellay and they were so rude and uncaring.	0.272
4	I hope such report which shows the end of the year profit/loss will be available soon; so that it can be easier for the users to file their tax declaration in the future.	0.278
5	It took me five weeks, yes FIVE WEEKS to get this sorted and my account back online and get access to my money again.	0.279

Table 18: Top 5 false negatives with the lowest classification probability

Amongst the false positives with the highest classification probabilities, all were originally clustered under cluster 14 (customer service – app experience). Evaluating the sentences, overlap between topics are apparent. The sentences continue to pertain towards aspects of customer service with little indication of separation. For instance, sentence number 2 refers closer to the app experience while the remaining sentences provide no indication that the experiences were encountered on the app. Conversely, with false negatives, the misclassification can only be tagged as the negative class (given it's a binary classifier). Regardless, through the context of the sentence, an approximation on the misclassification can be made. While sentence 4 potentially relates to business accounts, the remaining sentences refer to live chat experiences, lack of response and poor turnover times – aspects of customer service. Both represent misclassification but also presents the overlap of topics and opportunities for regrouping of clusters upon evaluation of the binary classification models over the remaining clusters.

Instead of hyperparameter tuning, the hyperparameter combination from Table 15 were utilised when developing binary random forest classification models for the remaining clusters. Given the resource intensity from k-fold cross validation and iterating over 27 clusters, the exclusion of hyperparameter tuning reduces the training time for each cluster at the expense of potentially suboptimal performances. Following the same sequence of arguments, the results over the 27 clusters were evaluated and are presented in Appendix 9.12.

False positives, negatives and topic similarities supported the regrouping of similar clusters into larger ones. Once the clusters are regrouped, classification models are refitted and revaluated, resulting in the final grouping of clusters. Examples of regrouped clusters are demonstrated in Table 19. Clusters with specific and niche topics (e.g. crypto and bank subscription plans) are kept as standalone clusters. Meanwhile, larger overarching clusters like customer service and account issues houses aspects of each topic. For example, unresolved ticket, live chat experience, and poor responses all pertain to customer service and hence, were grouped together under a larger customer service cluster. All regrouped clusters and model performances are shown in Appendix 9.13 and Appendix 9.14 respectively.

Final Cluster Number	Cluster Size	Old Regrouped Clusters	Assigned Topic

0	647	1 (Crypto)	Crypto
3	798	11 (Bank subscription plans)	Bank subscription plans
6	10367	14 (Customer service – app experience) 21 (Unresolved ticket) 30 (Live chat experience) 33 (Customer service – app experience) 34 (Poor responses) 35 (Customer service – non-app experience)	Customer service
7	3386	10 (Blocked account) 19 (Account opening) 39 (Blocked account) 42 (Account closure) 43 (Blocked account)	Account issues

Table 19: Final regrouped clusters example

Applying the series of trained models over the review sentences of the remaining banks, review sentences (which were unlabelled) are now automatically labelled to topics. For illustration, a review sentence from Wise is now assigned a series of topics (and not necessarily only a single topic). From the perspective of banks and other beneficiaries, this allows for internal monitoring of various services and/or offerings, and external benchmarking against the performance of its competitors. Figure 42 and Figure 43 provides an example to the following respectively.

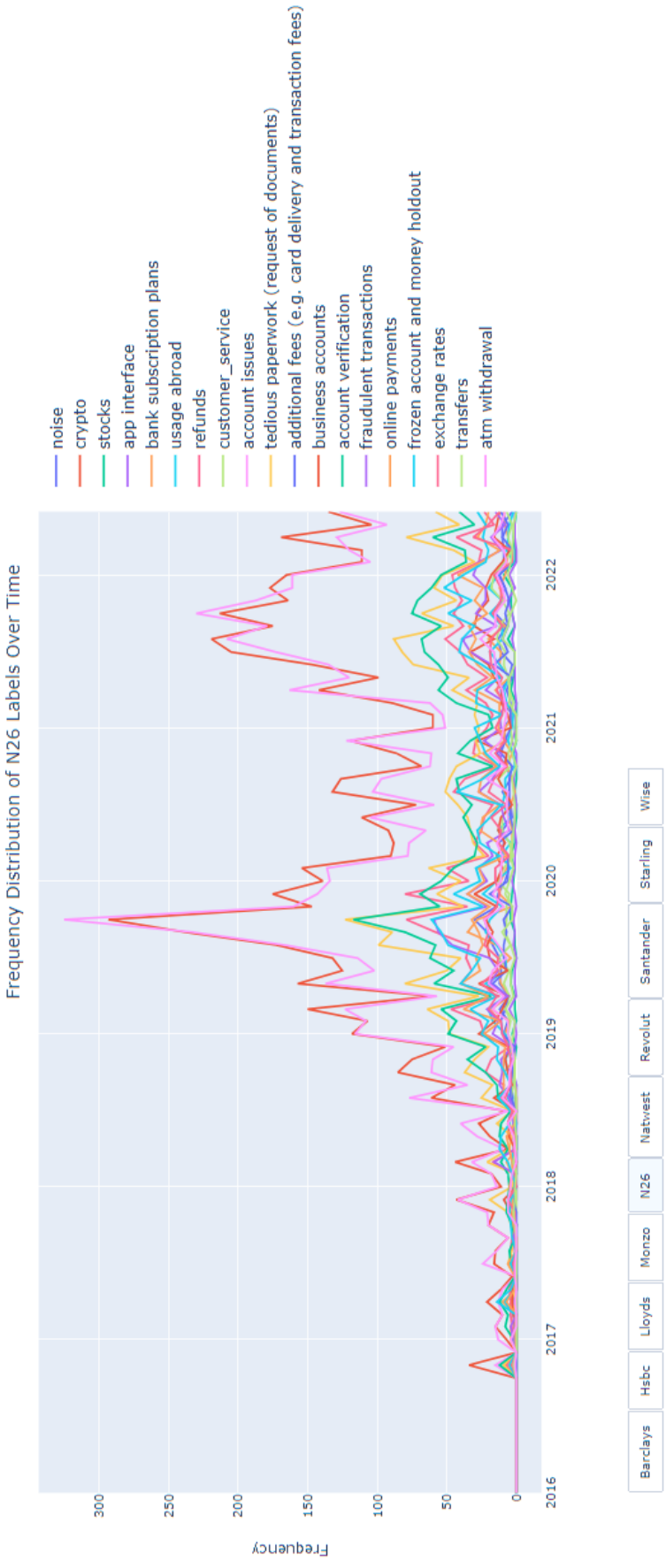


Figure 42: N26 frequency distribution by topic labels over time – internal monitoring

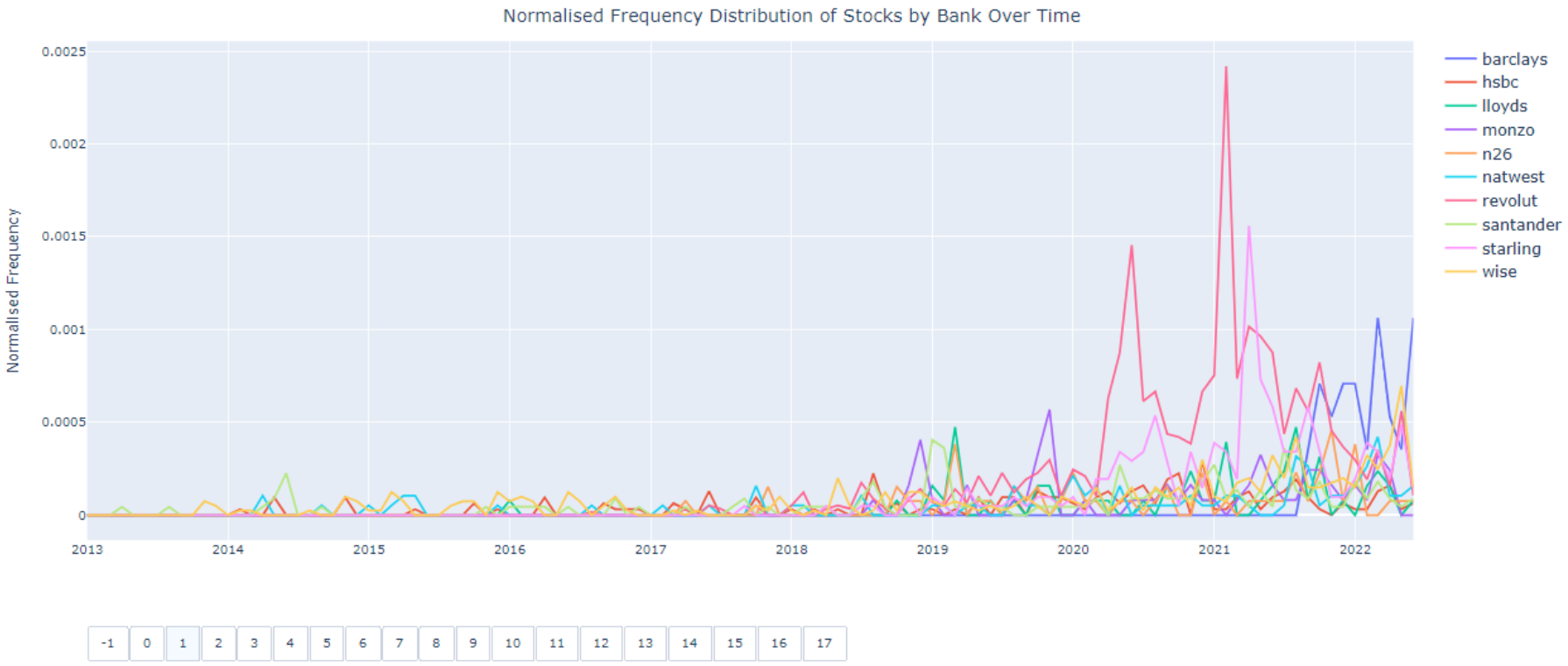


Figure 43: Normalised frequency distribution of stocks (cluster 1) by bank over time – benchmarking

Using N26 as an example to the former, a frequency distribution of the topic labels over time allows N26 to identify services and offerings which are performing poorer than others, allowing for prioritisation of improvements. Negative experiences surrounding crypto and account issues outweigh other functions across all years; signalling further investigation of both aspects as compared to other functions. Sharp increases (and subsequent decreases) shed light on prior challenges surrounding cryptocurrencies and account issues. Similarly, future challenges can be identified and acted upon.

With the latter, the frequency distributions were normalised by the total number of review sentences for the given bank. By comparing normalised frequencies, differences in number of review sentences are accounted for, allowing for fair comparisons in trend between banks. Benchmarking allows banks to compare its performance to its competitors. For example, Revolut's spike in negative experiences with stock signals to Revolut to take remedial actions (given that rate of poor experience exceeds its competitors). Conversely, it flags other banks to review their stock functions and potentially improve upon it to capitalise on weakened customer confidence.

4.5.2. Multi-label Classification

Despite the performance of the series of binary classifiers, scaling and applying it onto larger datasets may prove a challenge due to the computational intensity of deploying a series of models. Though this study utilises 19 binary classifiers, future works on differing datasets or domains may result in more binary classifiers. Thus, a single multi-label classifier is developed through transfer learning on a transformer-based model.

Prior to model fitting, the distribution of multi-labels is evaluated to assess the number of sentences by the number of labels. Large number of multi-tagged sentences with more than 2 labels dilutes the number of training samples as fewer sentences will fall under the same combination of labels. Figure 44 assesses it by plotting the normalised distribution (normalised by the total number of sentences) of multi-tagged sentences.

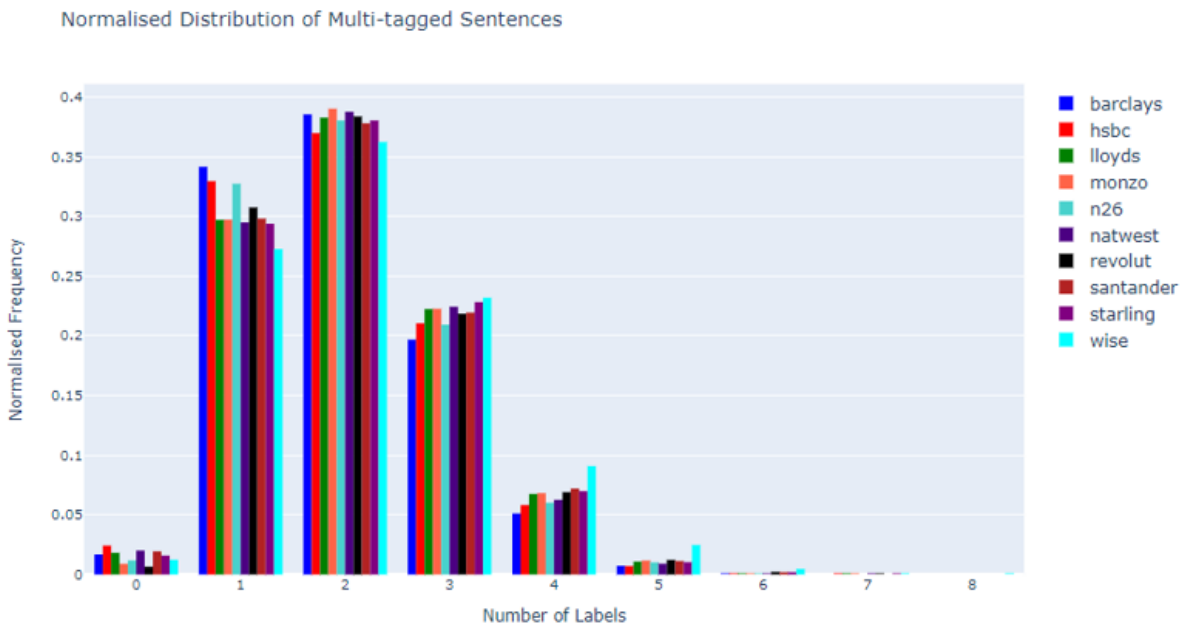


Figure 44: Normalised distribution of multi-tagged sentences

Sentences with 0 labels is a result of noise being a cluster of its own. Without a noise cluster, sentences not tagged with a label will be assigned as noise. The majority of sentences are tagged with 2 or fewer labels but around 25% of the sentences are tagged with more than 2 labels. Mitigating this, the number of labels was capped to an upper limit of 2 and the 2 labels with the highest classification probabilities was selected for sentences which originally had more than 2 labels. For example, a sentence tagged as refunds, customer service, and account issues, would take 2 of the tags with the highest classification probabilities. Even with 2 labels, there are 171 (19 x 18 ÷ 2) possible combinations of labels and the distribution of unique label combinations is assessed, as shown in Figure 45.

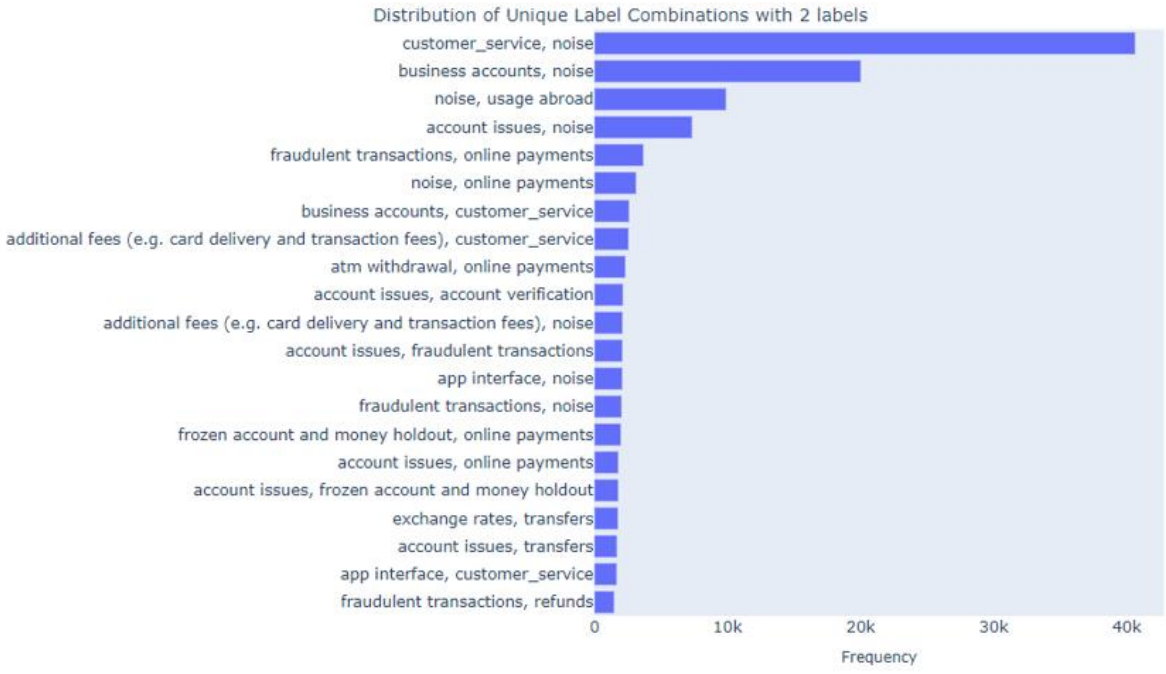


Figure 45: Distribution of unique label combinations with 2 labels (with noise)

A total of 170 unique 2-label combinations were identified with customer service and noise forming the large majority of it. The challenges with treating noise as a cluster is seen again and reinstated when identifying noise as a primary or secondary label to the top 4 most populated 2-label combinations. As mitigation, noise was removed (as a primary or secondary label) from 2-label combinations and the distribution was re-evaluated. Figure 46 and Figure 47 plots the distribution of unique 1 and 2-label combinations respectively.

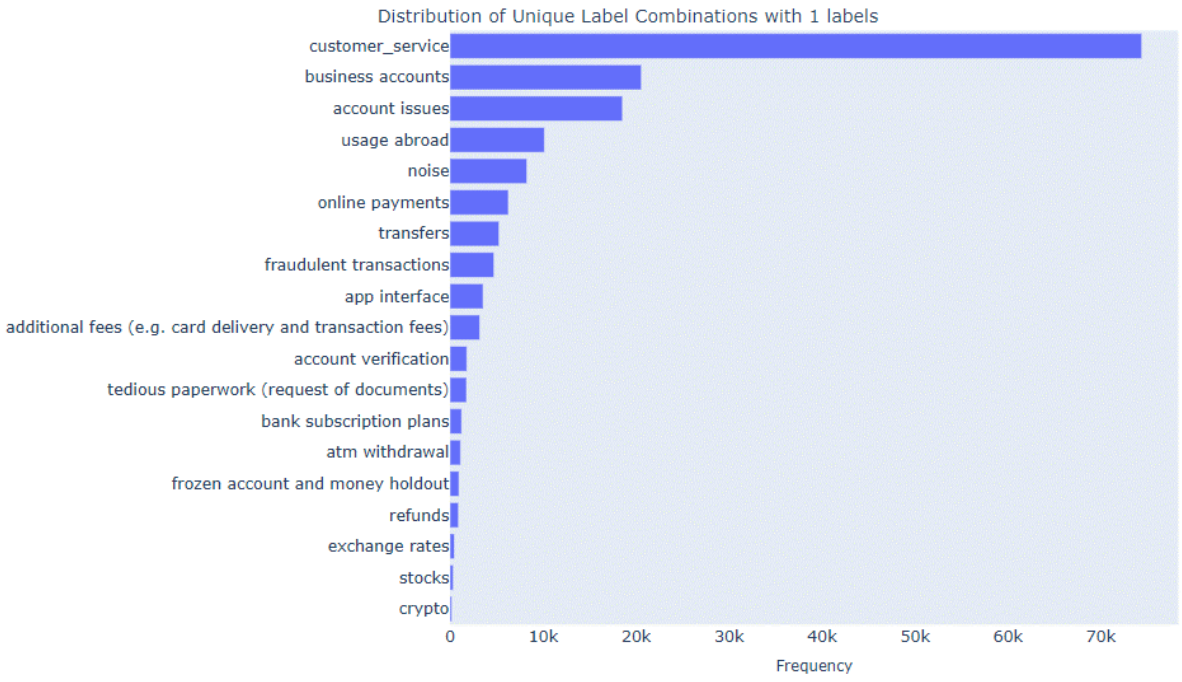


Figure 46: Distribution of unique label combinations with 1 label

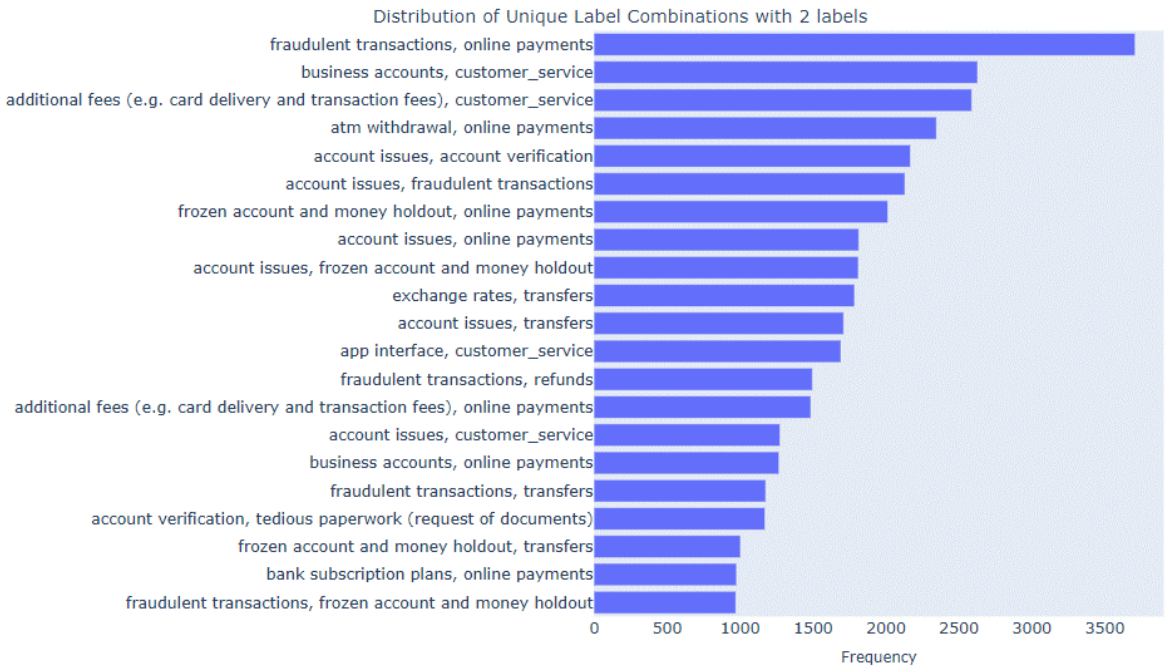


Figure 47: Distribution of unique label combinations with 2 labels (without noise)

With the removal of noise from 2-label combinations, more insightful label combinations can be identified – frequent customer service complaints and common combination of fraudulent transactions and online payments for example. Though, even through the exclusion of noise, a total of 152 unique 2-label combinations were identified. Lingering sparse label combinations may still be present and in particular, 43% (66) of unique 2-label combinations had frequencies of less than 100. Thus, a frequency threshold of 500 was adopted to account for the sparsity. Sentences with label combinations which fell below the frequency threshold were dropped, reducing the total number of sentences down to 217,865 from 233,699. The number of unique 1 and 2-label combinations were reduced down to 16 and 41 respectively as well.

Leveraging on a pre-trained BERT model, a multi-class multi-label classification model is trained over 80% of the sentences (and the remaining 20% for testing) through transfer learning. Populated labels like customer service and account issues were first down-sampled to minimise biases in predictions. Training it over 5 epochs, Figure 48 plots the losses and hamming score over epochs. Early stopping is implemented to terminate training if validation loss increased over two consecutive epochs.

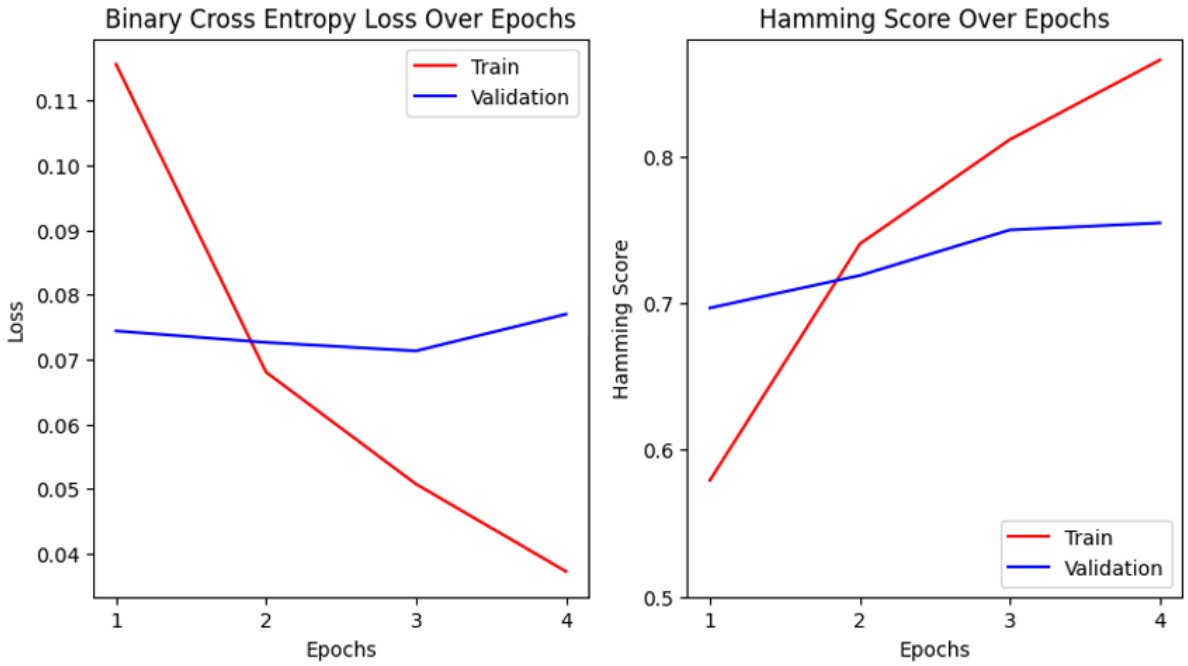


Figure 48: Binary cross entropy loss and hamming score over epochs

Learning is evident through the reducing losses over epochs. While training losses decreases steadily over epoch, validation loss decreases between epochs 1 to 3 before increasing in epochs 4 and 5 – signs of overfitting and prompting early stopping. The hamming score shares an identical narrative. While validation hamming scores increase at a diminishing rate and plateauing after epoch 3, training hamming scores increase with each epoch. Hamming score was used as a metric to deduce the model's performance since hamming score considers partially correct predictions (e.g. predicting label 3 correctly but label 10 wrongly), as compared to accuracy which considers fully correct predictions (Destercke, 2014). Given that the validation loss (0.073) was the lowest after 3 epochs of training, the model parameters after 3 epochs of training was selected for the final multi-class multi-label model. The corresponding validation hamming score was 0.719.

The misclassifications are evaluated through the precisions, recalls, and f1-scores (by label) and are presented in Table 20.

Label	Label - Topic	Number of Sentences	Metrics		
			Precision (%)	Recall (%)	F1-score (%)
-1	Noise	1324	62	27	38
0	Crypto	-	-	-	-
1	Stocks	-	-	-	-
2	App interface	841	92	84	88

3	Bank subscription plans	498	86	48	61
4	Usage abroad	2087	76	72	74
5	Refunds	527	64	86	73
6	Customer service	4881	83	82	82
7	Account issues	4944	94	85	89
8	Tedious paperwork (request of documents)	756	77	82	79
9	Additional fees (e.g. card delivery and transaction fees)	1263	81	54	65
10	Business accounts	4257	92	64	75
11	Account verification	1029	88	68	77
12	Fraudulent transactions	2630	73	87	79
13	Online payments	3497	82	83	83
14	Frozen account and money holdout	1167	71	73	72
15	Exchange rates	431	95	48	64
16	Transfers	2096	89	90	90
17	ATM withdrawal	1092	82	54	65

Table 20: BERT model performance over validation set

The recall of most labels (excluding label 5 and 12) fell below its recall, suggesting higher rates of false negatives compared to false positives – indicating preferences in prediction towards the positive class over the negative class. As a result, sentences tend to be over-labelled with more labels than its true set of labels – potentially due to non-mutually exclusive topic labels. Within the domain, topics are often referenced in conjunction (e.g. a business account opening issue) and more granular clusters may have been beneficial – a point of consideration for future works.

Larger labels (e.g. customer service, account issues, online payments and transfers) tended to outperform smaller labels, likely due to the larger number of samples to learn over. Contrastingly, classifications over smaller yet specific topics, like app interface, were accurate. Niche topics tend to be exclusive with clear distinctions from other topics; further emphasising the importance of clustering and sentence quality. Crypto and stocks were removed from the dataset as a result of implementing the frequency threshold and hence, sentences were not tagged as either labels.

The model was applied onto an unseen testing set (made up of 20% of the dataset) and the final results are shown in Table 21. The performance of the model is reflective of quality of the preceding

components within the pipeline and is caveated by the compounding errors of it as well. The learnings and insights obtained from the evaluation of each component provide inspiration and corrections for future works.

Dataset	Loss	Hamming Score
Testing	0.068	0.751

Table 21: BERT model performance over testing set

5. Discussion

The findings and approaches within this study produced results that, though can be improved upon, provides promise and prospect in the use of alternative unlabelled data that is readily available – supplying advancements to potential insights that companies (and other beneficiaries) can obtain and developments to the NLP space as a whole.

Reflecting upon our objectives, we showcased the capabilities of web-scraping in generating a robust and extensive dataset formed from customer banking reviews left on Trustpilot. We advertised the vast amount of unstructured (and semi-unstructured) data by scraping close to 300,000 reviews left on challenger and conventional banks. We discussed the ethics surrounding web-scraping and though our approaches leveraged upon a user's username to approximate their gender, it served as an exploratory exercise to compare the demographics of users between banks. Our feature extraction and modelling tasks took only the reviews and did not include any idiosyncratic factors (e.g. gender or age) that may discriminate or bias against a certain demographic of users. We deemed (and still deem) ethics to be preserved and reiterate the application of the web-scraped dataset as an academic exercise to produce insights over an unlabelled dataset. The use of download delays was also adopted to relieve traffic from Trustpilot's web servers.

Through EDA, we presented evidence of differing structural behaviours between challenger and conventional banks – we saw significant differences in two areas: size of online presence and composition of reviews. We proved the differing online footprints between challenger and conventional banks with challenger banks having almost 16 times more reviews altogether and it being made up predominantly of good reviews (as compared to poor reviews for conventional banks). Supplementing it, we found challenger banks to interact with reviews more than conventional banks but both place priority on attending to (proven) longer poor reviews first compared to shorter good reviews. Using unigrams and bigrams, we identified stop words to aid the data cleaning process and preliminary topics – providing promise to succeeding components but caveated by considerations on the extensiveness of the dataset and generalisability of models onto both bank types. In hindsight, further analysis on misclassifications by bank type may have proven useful to address this.

We displayed the use of unsupervised learning techniques in sequence to generate a taxonomy of topics within the banking domain – customer service and exchange rates for example. We demonstrated the strengths of word embeddings as a way to bring text-based data onto a numeric space whilst preserving semantics and relationships between words and sentences, and experimented over two different embeddings models. Dimensionality reduction through UMAP justified its use to allow for visual analysis on an otherwise high dimensional representation, and clustering via HDBSCAN proved integral in grouping similar sentences together. Through topic modelling techniques, we derived topics for each cluster – forming a taxonomy of topics within the banking domain without manually assigned

labels. Though one may argue that regrouping of clusters required manual intervention, the final cluster groupings and topics were derived from initial generated topics. This exercise emphasised the importance of expert judgement and existing challenges with evaluating unsupervised learning techniques, even when employing metrics like DBCV.

Over a subset of clustered sentences (Revolut), we developed a series of binary classifiers for each cluster, achieving excellent validation (and training) accuracy scores over all clusters with the lowest accuracy (excluding the noise cluster) being 85%. We reiterated the importance of experimenting and evaluation in improving cluster quality and overall model performance by displaying the series of iterative results prior to arriving at the final set of clusters – including the significance of false positives and negatives. We illustrated problems on treating noise as a cluster (and developing a binary classifier for it) and would have excluded it as a modelled cluster label in retrospect. Extending it over the entire dataset, we generated multi-labels for each sentence, forming a multi-labelled dataset. We discussed the appropriateness of Revolut as the training set used for binary classifier model development since it had an extensive list of services but upon review, assessing the misclassifications by bank types would have helped to validate this assumption.

With a multi-labelled dataset, we were able to employ supervised learning techniques to classify sentences into a series of labels and topics. We reviewed and showed the flexibility of transformer-based approaches like BERT by fine-tuning the model architecture to our given task. Specifically, we exhibited that by adding hidden layers to the BERT architecture, the outputs to the BERT model can be mapped to a series of labels. From it, we presented reliable text classification results, achieving a hamming score of 0.751 over an unseen set of sentences.

We illustrated the significance of hyperparameter tuning, implications of different hyperparameters, and the value in experimentation on the overall performance. As part of implementation of unsupervised learning techniques, we tuned hyperparameters in tandem and toyed with varying sentence input structures to find the optimal results given the constraints of this study. We found certain data pre-processing steps to actually hinder the quality of embeddings and in hindsight, would have led to time savings by simplifying the data pre-processing steps but served as a valuable learning exercise for future works regardless. We highlighted grid search as a tool to aid hyperparameter tuning when a single performance metric can be defined but only performed it over a single binary classifier. Additionally, we did not experiment with alternative architectures or hyperparameter values when implementing the BERT model. While it is unclear how much the model would have improved by, we had to weigh the trade-offs of a resource intensive process against (potential) model improvements and due to resource constraints, we chose against the latter. However, with learnings from this project, future iterations and time savings can allow for reallocation of resources for extensive experimentation on binary and multi-

label classifiers. Besides, experimentation and hyperparameter tuning can only ever improve the results and at worse keep it the same.

By aggregating the predictions of our models at the topic and bank-level, we demonstrated how companies are able to draw a diverse set of insights. Through a timeseries representation, we showed how aggregation at the topic-level allows for companies to benchmark against its competitors – citing Revolut's battle with stock-related services and indicating potential capitalisation opportunities (e.g. improvements and/or marketing) for other banks. Contrastingly, by aggregating predictions at the bank-level, we displayed means for companies to evaluate its own services in relation to their ecosystem of services. As an example, we depicted N26's large negative sentiment surrounding cryptocurrencies, account issues (in relation to its other services), and proposed leveraging upon these insights to better prioritise innovation, improvement, and recovery strategies.

Overall, we found great success in our objectives presented in Section 1. We summarised how each objective was achieved and even with those that fell slightly short (e.g. experimentation), we provided critical analysis on the challenges and improvements for future iterations. Reflecting upon our research question proposed: "**Can an NLP-based approach on entirely unlabelled data derive insights on potential trends and improvements to a company's offering and services?**", insights on a company's service and offerings can, in-fact, be derived through an NLP-based approach - even on an entirely unlabelled dataset. We believe our results and findings show that this study made good attempts at utilising unstructured data and unsupervised, semi-supervised, and supervised learning techniques to ultimately generate insights. We believe that our study serves a good by-product of the learnings from our literature review with real-world applications. Though our study leaves off with improvements to be made, we believe that this study supplied anecdotal evidence to form a strong basis for future works. In short, to answer our research question in a few words: we can.

6. Evaluation, Reflections and Conclusions

As seen in Section 5, we discussed the successes, shortcomings of our objectives, and how it provided clear stepping stones towards answering our research question. Overall, the choice of objectives was both appropriate and sufficient in the context of this study but diverged slightly from the original set of objectives in our project proposal. Not only did the final set of objectives account for the added scope of multi-label classification, it provided clearer checkpoints to evaluate the progress of the study and inch us closer towards answering our research question.

At face value, the objectives seem simple but we found the objectives to be challenging and demanding at times, albeit fitting and thought provoking – especially when working with techniques that was once new and unfamiliar; requiring to lend heavily from the literature. Producing our dataset proved to be a challenge given the structure of HyperText Markup Language (HTML) components within Trustpilot. For example, similar information like "Date" and "Edit Date" were encapsulated under entirely distinct sets of elements. Web scraping was also precarious when balancing the speed at which reviews were scraped and the preservation of ethics. Challenges around clustering (and unsupervised learning) evaluation was seen again within this study even when employing measures like DBCV, leaving potential debates on optimal clustering. Furthermore, the resource intensive process of model development (binary and multi-label) led to minimal experimentation surrounding it. Regardless, despite the challenges, we still found several successes across the list of objectives and the objective choices are deemed apt within the context of this study.

By working towards our objectives, we have implemented a range of unsupervised, semi-supervised, supervised learning techniques, throughout this study and produced findings from it. Through EDA, we found structural differences between the online presence conventional and challenger banks in terms of both, number of reviews left by customers and interactivity with replies. We found conventional banks to have significantly fewer reviews and interacts with (close to) none through replies. Conversely, challenger banks frequently replied to poor reviews and garner almost 16 times more reviews. We learned and were reminded of the importance of EDA on exploring underlying trends within the data and its strengths to produce considerations that may impede upon the relevance of our results.

By generating word embeddings, reducing it to a lower representation, clustering, and assigning topics onto each cluster, we found it possible to automatically assign topic labels to an otherwise unlabelled set of sentences. From it, we found most complaints to surround customer service across both banks (even conventional banks which often have physical branches). Additionally, we found data cleaning steps within data pre-processing to be redundant when employing a transformer-based word embedding model. We learned about the significance of hyperparameter tuning on clustering quality (and model performance) and the value in having a deep understanding of our dataset and domain.

Additionally, the value of expert judgement was emphasised especially in the context of evaluating clustering quality and opportunities to improve it.

Through semi-supervised learning over a subset of (now) labelled sentences, we found pairs of topics which are commonly associated together (e.g. fraudulent transactions and online payments) and extending the predictions over the entire dataset, we found that historic insights were indeed able to be generated. We learned about the use of down-sampling to minimise biases in predictions and the relevance of classification probabilities, false positives, and negatives in evaluating classification and clustering quality. We saw the latter help to identify cluster regrouping opportunities. Moreover, we learned about BERT as an approach to condense a series of binary classifiers into a single multi-label classifier and how we can fine tune the model architecture to the context of text classification.

The literatures reviewed were suitable and lent itself highly to this study. Each literature supplied us with novel NLP-related approaches and provided a temporal narrative to the developments within each component. Through it, we learned about the shortcomings of preceding approaches which helped to aid our approach design process and implement state-of-the-art approaches like UMAP, HDBSCAN, and BERT, to solve our research question. While the literature still had to be tailored to the context of this study (e.g. domain-specific considerations and hyperparameter tuning), it formed a sturdy and reliable foundation. The literatures from the proposal carried nicely into the actual project but upon review, additional literature surrounding model interpretability may be useful for future work and to users of our work products. For example, key performance indicators (KPI) and decisioning (e.g. reprioritisation and re-strategising) within companies often require auditability, interpretability, and explainability which black-box approaches (like BERT) find difficult to provide.

Aside from the main beneficiaries highlighted in Section 1, our study, results, and findings contribute to the NLP space as a whole. Not only have we provided an additional application example to a suite of NLP-based techniques, we also showcased the use of unstructured data despite having no labels. Though legal and ethical considerations need to be made if the data is scraped, our study may hopefully inspire solutions for data collection and view unstructured data as a resource instead of an impotence. This is particularly useful for NLP-related tasks which works over a niche domain or domains where structured data is scarce. BERT-based solutions on an array of tasks likely exist but we showed potential real-world use case for the results that benefit companies and institutions.

Throughout the project, the work plan presented in the project proposal was frequently referenced – ensuring that required amounts of time sufficient time are available for different components and the completion of this project respectively. It however, had to be adapted slightly to account for the added scope of multi-label classification – allocating a month after topic classification model development (revised to text classification). Topic classification model development was revised to text classification and evaluation was performed within each component. Regardless, aside from the revisions, the work

plan was abided to with minimal deviations and all milestones were reached but feature extraction and multi-label classification demanded more time than expected – taking between 1 to 2 weeks more. However, due to an initial conservative working plan, the additional required time was allocated by reducing the amount of time spent on report writing. Though, in hindsight, now equipped with the knowledge that data-cleaning steps proved redundant, less time would have been rationed to data pre-processing.

In addition, initial (and now corrected) inefficient implementation led to additional time loss and resource wastage which in retrospect, can be implemented directly in future works. For example, initial implementation of hyperparameter tuning over dimensionality reduction and clustering were initially both performed over each hyperparameter combination (thus performing both over 27 combinations). However, the hyperparameters for clustering has no implications on the quality of low-dimensional representation (but the converse is not true). Upon re-implementation, low-dimensional representations over 9 combinations were first obtained prior to being used as inputs to 3 varying clustering hyperparameter values – still resulting in 27 total combinations. Furthermore, the noise cluster provided challenges by frequently serving as primary or secondary labels to multi-labelled sentences and misclassification. Though remedial steps were taken (i.e. dropping noise from 2-label combinations), future implementations may benefit from excluding noise from the list of clusters. Conversely, implementing a frequency threshold to limit the number of unique label combinations proved effective but further experimentation (evaluated through number of unique label combinations and impact on model performance) with varying frequency thresholds may prove beneficial.

Aside from time and resource savings, this study offers room for future works to (potentially) improve the quality of results and form a more extensive study. For instance, performing hyperparameter tuning on each fitted binary and multi-label classification model may prove. While the hyperparameters for random forests were defined, hyperparameters and architecture design for our BERT model are candidates for variation. For example, toying with varying batch sizes, number of hidden layers, and learning rate, are all an initial set of hyperparameters to experiment with which can ultimately improve model performance, mitigate overfitting, and garner more detailed insights.

While insights on poor reviews supply evidence for poor performing services, good reviews provide context to the overall performance – allowing for fairer conclusions to be made. For example, consider if poor customer service and exchange rates were shown through 1000 and 200 sentences respectively, then looking at poor reviews alone will conclude that customer service is performing far worse. However, if good customer service and exchange rates were shown through 10000 and 500 sentences respectively, then a more normalised view (e.g. proportion of poor reviews) on the performances can be obtained. Here, it is clear that exchange rates will require more focus. Besides, good reviews also supply companies with explicit views on good practices by its competitors – providing opportunities to

learn from. To implement this for future works, we propose repeating the feature extraction and text classification process over good reviews instead (along with improvements from discussions and evaluations).

Currently, the insights drawn are at the sentence-level rather than the review-level due to the tokenisation of reviews into sentences and subsequent processing at the sentence-level. As a more accurate depiction and representation, review-level insights may be more reflective. For example, consider a long review regarding customer service and a short review regarding exchange rates made up of 10 sentences and 1 sentence respectively. Assuming that each sentence is mono-topic (i.e. surrounding a single topic), then poor experiences around customer reviews are amplified and depicted as a larger priority than addressing exchange rate issues. Though this provides an extreme case and averaging may occur, exploring ways to aggregate sentences back to its constituent review can account for it. Though it is currently unclear on how this may be done, taking the arguments of the maxima (argmax) on the normalised sum of classification probabilities may be one such approach.

As a final improvement, we observed that the insights drawn are currently a historic or incurring view, rather than a forward-looking one. In reality, changes in business strategy is costly and requires time to design, be approved, and implement. Thus, it is naïve to assume rapid response by companies. A forward-looking approach would be ideal to get predictions on the influx of good and poor reviews, though reviews are mainly dependent on performance and quality of services which makes it hard to model. Time could be a factor of consideration to capture any cyclicality in reviews (e.g. more stock related reviews after financial year ends) but may not amount to much for topics which are independent of time. Instead, we suggest adding to the ecosystem, for future works, scraping capabilities to perform regular interval scraping (e.g. daily or weekly). Running it through the pipeline, topic predictions can be obtained such that companies or users are constantly up-to-date with review trends.

7. Glossary

Activation Function	The function in a neuron which determines the output of the node from a given set of inputs – dictating whether it is activated or not.
Backpropagation	A process involved when training a neural network, whereby weights are fine-tuned through a backward pass on the evaluation of the errors of the forward pass.
Backward Pass	The process of traversing through all nodes from output to input layer, in reverse order.
Challenger Banks	Online and tech-based retail banks that rival the financial product offerings of conventional banks through its digital solutions.
Consumer Banking	Equivalent of retail banking, whereby financial services are provided to individual consumers instead of businesses.
Conventional Banks	Fully-fledged banks which operate with physical branches, offering traditional services like deposits and loans.
Cross Entropy Loss	A measure of difference between two probability distributions, p and q, measuring the number of bits needed to encode data from distribution p using distribution q. Generally, lower cross entropy losses are associated to better model performance.
F1-score	The harmonic mean of precision and recall.
Final Layer	Equivalent to the output layer which produces predictions from a set of weighted inputs passed through an activation function.
Forward Pass	The process of traversing through all nodes from input to output layer, in order.

Grid Search	A process which exhaustively searches the hyperparameter space for optimal combinations in accordance to a targeted outcome or metric.
Hidden Layer	A sequence of nodes that fall between input and output layers which produces an output generated by parsing a set of weighted inputs through an activation function.
Hyperparameter	A parameter (or set of parameters) that is extrinsic to models and/or techniques which affects the learning and performance.
Loss Function	A function which measures the quality of predictions by comparing it to target values and is aimed to be minimised.
One-hot Encoded Vectors	A binary representation for categorical variables.
Pipeline	A series and sequence of processes which converts raw data into products and insights by parsing the output of one process as the input to the subsequent process.
Precision	The proportion of correctly labelled positive class amongst positive class predictions. The lower the precision, the higher the misclassification of the negative class.
Recall	The proportion of correctly labelled positive class amongst labels whose ground truth is the positive class. The lower the recall, the higher the misclassification of the positive class.
Rectified Linear Unit	A piecewise linear activation function which outputs the input directly if it is positive and 0 if it is negative.
Retail Arm	A consumer banking practice within banks.

Semi-supervised Learning	The training and learning approach which involves a combination of small number of labelled samples and large number of unlabelled samples.
Sigmoid	An activation function which takes real values as input and outputs values between 0 to 1. Larger input signals move the output closer to 1 while smaller output signals move the output closer to 0.
Softmax	A function which produces a probability distribution in accordance to the magnitude of values in a vector.
Supervised Learning	The training of models and algorithms using labelled datasets.
Unsupervised Learning	The use of machine learning algorithms or methods to identify patterns and trends within an unlabelled dataset.

8. References

Alashwal, H. *et al.* (2019) 'The Application of Unsupervised Clustering Methods to Alzheimer's Disease', *Frontiers in Computational Neuroscience*, 13.

Aytekin, C. (2022) 'Neural Networks are Decision Trees'.

Beel, J. *et al.* (2016) 'Research-paper recommender systems: a literature survey', *International Journal on Digital Libraries*, 17(4), pp. 305–338.

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) 'Latent Dirichlet Allocation', *The Journal of Machine Learning Research*, 3(null), pp. 993–1022.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32.

Calders, T. et al. (eds) (2014) Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science).

Campello, R.J.G.B., Moulavi, D. and Sander, J. (2013) 'Density-Based Clustering Based on Hierarchical Density Estimates', in J. Pei et al. (eds) *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 160–172.

Caruana, R. and Niculescu-Mizil, A. (2006) 'An empirical comparison of supervised learning algorithms', in *Proceedings of the 23rd international conference on Machine learning - ICML '06*. *the 23rd international conference*, Pittsburgh, Pennsylvania: ACM Press, pp. 161–168.

Cer, D. *et al.* (2018) 'Universal Sentence Encoder', p. 7.

Confidence to Deploy AI with World-Class Training Data (no date) *Appen*. Available at: https://stage.appen.com/ (Accessed: 15 December 2022).

Destercke, S. (2014) 'Multilabel Prediction with Probability Sets: The Hamming Loss Case.', in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2014)*. France, France, pp. 496–505.

Devlin, J. *et al.* (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'.

Dudoit, S. and Fridlyand, J. (2002) 'A prediction-based resampling method for estimating the number of clusters in a dataset', *Genome Biology*, 3(7), p. research0036.1.

Edunov, S. *et al.* (2018) 'Understanding Back-Translation at Scale', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. *EMNLP 2018*, Brussels, Belgium: Association for Computational Linguistics, pp. 489–500.

GitHub - lmcinnes/umap: Uniform Manifold Approximation and Projection (no date) *GitHub*. Available at: https://github.com/lmcinnes/umap (Accessed: 1 December 2022).

González-Carvajal, S. and Garrido-Merchán, E.C. (2021) 'Comparing BERT against traditional machine learning text classification'.

hdbscan/how_hdbscan_works.rst at master · scikit-learn-contrib/hdbscan (no date) *GitHub*. Available at: https://github.com/scikit-learn-contrib/hdbscan (Accessed: 5 December 2022).

Howard, J. and Gugger, S. (2020) 'fastai: A Layered API for Deep Learning', *Information*, 11(2), p. 108.

Kowsari, K. *et al.* (2019) 'Text Classification Algorithms: A Survey', *Information (Switzerland)*, 10.

Khurana, D. *et al.* (2022) 'Natural language processing: state of the art, current trends and challenges', *Multimedia Tools and Applications* [Preprint].

Krotov, V. and Silva, L. (2018) 'Legality and Ethics of Web Scraping', in. *Americas Conference on Information Systems*.

Kuang, D., Choo, J. and Park, H. (2015) 'Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering', in M.E. Celebi (ed.) *Partitional Clustering Algorithms*. Cham: Springer International Publishing,

Landauer, T.K., Foltz, P.W. and Laham, D. (1998) 'An introduction to latent semantic analysis', *Discourse Processes*, 25(2–3), pp. 259–284.

Lee, D. and Seung, H.S. (2000) 'Algorithms for Non-negative Matrix Factorization', in *Advances in Neural Information Processing Systems*. MIT Press.

Liu, P., Qiu, X. and Huang, X. (2016) 'Recurrent Neural Network for Text Classification with Multi-Task Learning'.

van der Maaten, L. and Hinton, G. (2008) 'Viualizing data using t-SNE', *Journal of Machine Learning Research*, 9, pp. 2579–2605.

MacQueen, J. (1967) 'Some methods for classification and analysis of multivariate observations', *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 5.1, pp. 281–298.

Mandelbaum, A. and Shalev, A. (2016) 'Word Embeddings and Their Use In Sentence Classification Tasks'.

McInnes, L. and Healy, J. (2017) 'Accelerated Hierarchical Density Clustering', in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 33–42.

McInnes, L., Healy, J. and Melville, J. (2020) 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'.

Moulavi, D. et al. (2014) 'Density-Based Clustering Validation', in Proceedings of the 2014 SIAM International Conference on Data Mining. Proceedings of the 2014 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, pp. 839–847.

Nam, J. *et al.* (2014) 'Large-Scale Multi-label Text Classification — Revisiting Neural Networks', in T. Calders et al. (eds) *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 437–452.

NatWest fined £264.8 million for anti-money laundering failures (2021) *FCA*. Available at: https://www.fca.org.uk/news/press-releases/natwest-fined-264.8million-anti-money-laundering-failures (Accessed: 15 December 2022).

New Research Report: Global Customer Loyalty a $323B Endeavor (2019). Available at: https://www.businesswire.com/news/home/20190503005033/en/New-Research-Report-Global-Customer-Loyalty-a-323B-Endeavor (Accessed: 15 December 2022).

Pennington, J., Socher, R. and Manning, C. (2014) 'Glove: Global Vectors for Word Representation', in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543.

Rajaraman, A. and Ullman, J.D. (eds) (2011) 'Data Mining', in *Mining of Massive Datasets*. Cambridge: Cambridge University Press, pp. 1–17.

Sander, J. (2010) 'Density-Based Clustering', in C. Sammut and G.I. Webb (eds) *Encyclopedia of Machine Learning*. Boston, MA: Springer US, pp. 270–273.

'scrapy/scrapy' (2022). Scrapy project. Available at: https://github.com/scrapy/scrapy (Accessed: 7 November 2022).

Sorzano, C.O.S., Vargas, J. and Montano, A.P. (2014) 'A survey of dimensionality reduction techniques'.

spacy/en_core_web_trf · Hugging Face (no date). Available at: https://huggingface.co/spacy/en_core_web_trf (Accessed: 15 December 2022).

TensorFlow Hub (no date a). Available at: https://tfhub.dev/google/universal-sentence-encoder/4 (Accessed: 9 December 2022).

TensorFlow Hub (no date b). Available at: https://tfhub.dev/google/universal-sentence-encoder-large/5 (Accessed: 9 December 2022).

Thalamuthu, A. *et al.* (2006) 'Evaluation and comparison of gene clustering methods in microarray analysis', *Bioinformatics*, 22(19), pp. 2405–2412.

Vaswani, A. *et al.* (2017) 'Attention is All you Need', in *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Wang, S., Huang, M. and Deng, Z. (2018) 'Densely Connected CNN with Multi-scale Feature Attention for Text Classification', in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. *Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18}*, Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, pp. 4468–4474.

Who Delivers More Value—Banks or Technology Companies? (2018) *Bain*. Available at: https://www.bain.com/insights/who-delivers-more-value-banks-or-technology-companies-snap-chart/ (Accessed: 15 December 2022).

Wiese, G., Weissenborn, D. and Neves, M. (2017) 'Neural Domain Adaptation for Biomedical Question Answering'.

Zhou, C. *et al.* (2015) 'A C-LSTM Neural Network for Text Classification'.

9. Appendices

9.1. Appendix A – Resource Dependencies

Component	Server	Number of Central Processing Units Used	Number of Graphics Processing Units Used	Memory (GiB)
Data Collection	Local	4	0	8
Data Pre-Processing and Engineering	Virtual	16		64
Exploratory Data Analysis				
Feature Extraction		96		384
Text Classification - Binary				
Text Classification – Multi-label		8	1	32

9.2. Appendix B – Web-scraping Settings

Setting Parameter	Value	Description
ROBOTSTXT_OBEY	False	Allows the spider to bypass scraping policies set out by websites
DOWNLOAD_DELAY	random.random()	Takes a random amount of time between 0 and 1 seconds before downloading consecutive pages, ensuring manageable server traffic
AUTOTHROTTLE_ENABLED	True	Finds the optimal crawling speed between the initial and maximum download delay
AUTOTHROTTLE_START_DELAY	1	Initial download delay in seconds
AUTOTHROTTLE_MAX_DELAY	10	Maximum download delay in seconds

9.3. Appendix C – List of Contractions and Full-forms

Contraction	Full-form
what’s	what is
wasn’t	was not
can’t	can not
it’s	it is
don’t	do not
doesn’t	does not
i’m	i am
didn’t	did not
hasn’t	has not
haven’t	have not
couldn’t	could not
shouldn’t	should not
won’t	will not
wouldn’t	would not
mustn’t	must not
shan’t	shall not
weren’t	were not

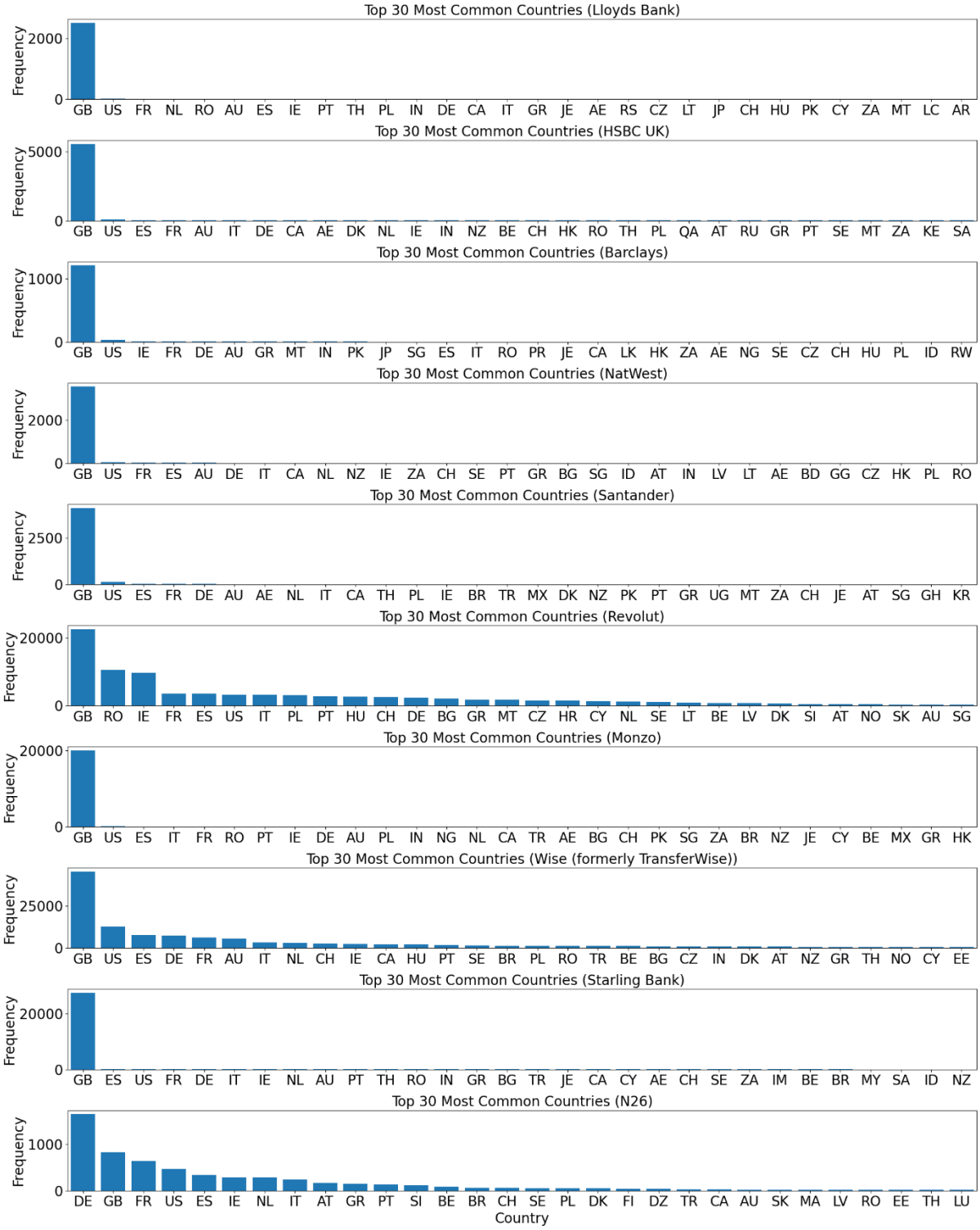
9.4. Appendix D – List of Stop Words

Type	Words
Across all banks	'them', 'would', 'now', 'come', 'who', 'had', 'all', 'easy', 'card', 'her', 'this', 'seem', 'they', 'an', 'set', 'his', 'use', 'so', 'am', 'put', 'if', 'their', 'been', 'my', 'banking', 'go', 'say', 'with', 'want', 'have', 'can', 'are', 'on', 'and', 'be', 'see', 'for', 'then', 'need', 'money', 'make', 'ask', 'that', 'at', 'get', 'the', 'no', 'is', 'do', 'i', 'we', 'know', 'were', 'she', 'which', 'or', 'from', 'a', 'banks', 'he', 'try', 'think', 'by', 'was', 'as', 'when', 'in', 'of', 'bank', 'it', 'will', 'your', 'what', 'but', 'has', 'to', 'you', 'us', 'me', 'did'
Bank-specific	Bank’s name (e.g. ‘Monzo’ would be treated as a stop word and removed from Monzo reviews but not from Revolut reviews)

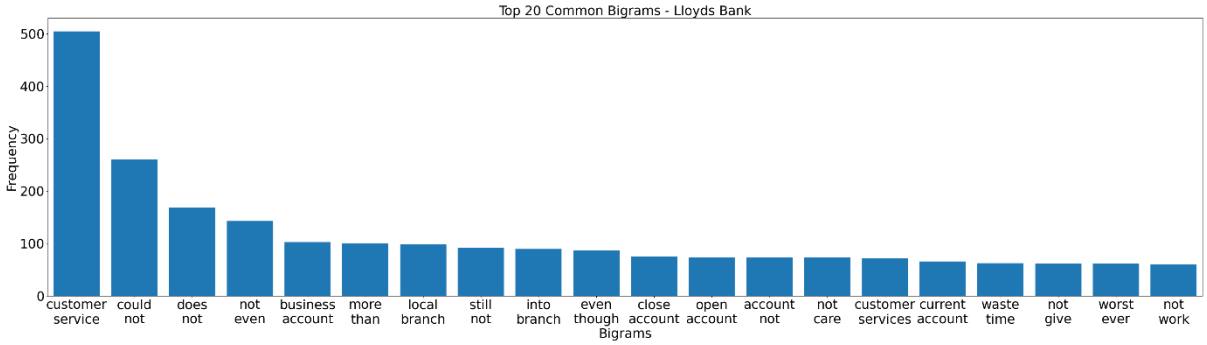
9.5. Appendix E – List of Punctuation and Characters

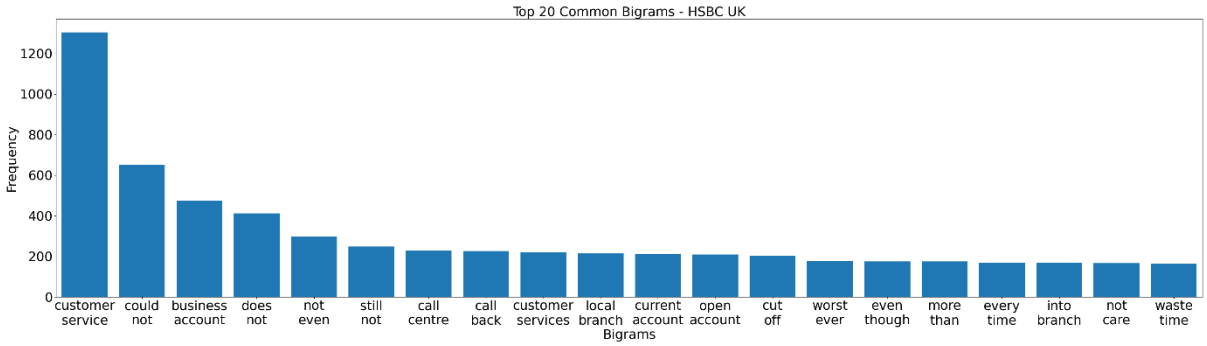
Punctuation and Character Marks	Treatment
“”, ““”, “,”, “/”, “`”, “\r\n”, “<”, “>”, “[”, “]”, “(”, “)”, “{”, “}”, “*”, “-”, “^”	Removed
“!”, “.”, “?”	Kept

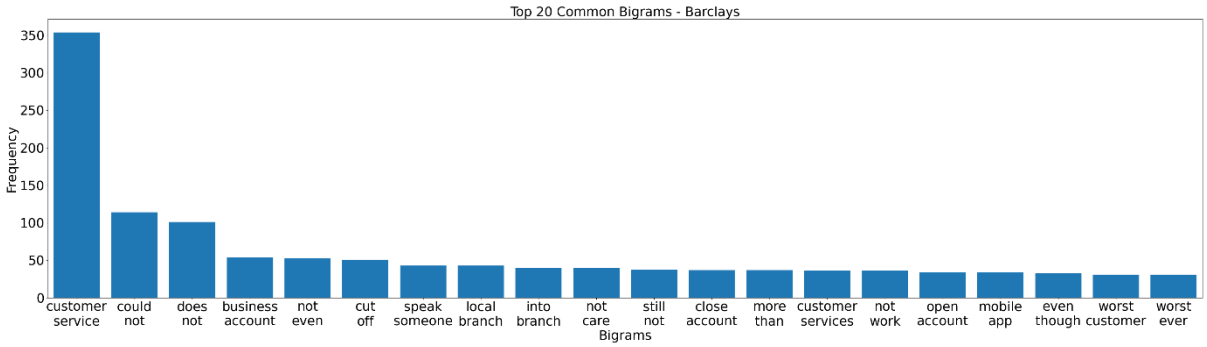
9.6. Appendix F – Distribution of User Location by Bank

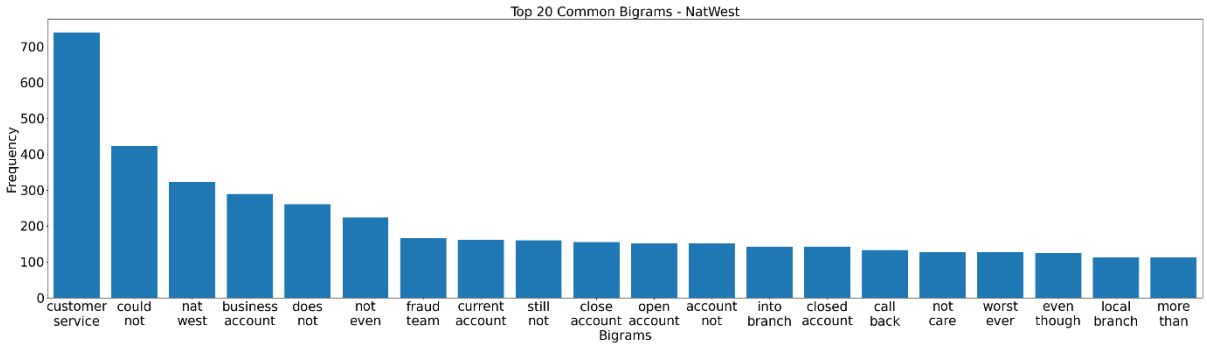


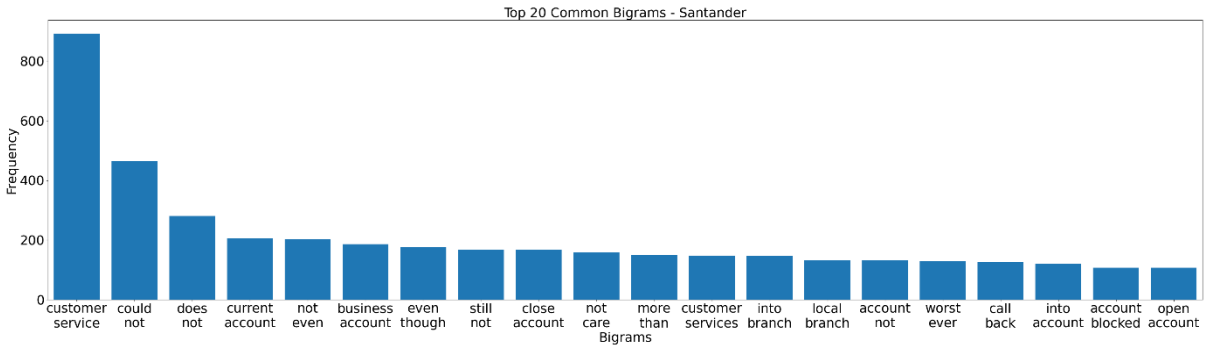
9.7. Appendix G – Bigram Distribution by Bank After Stop Word Removal

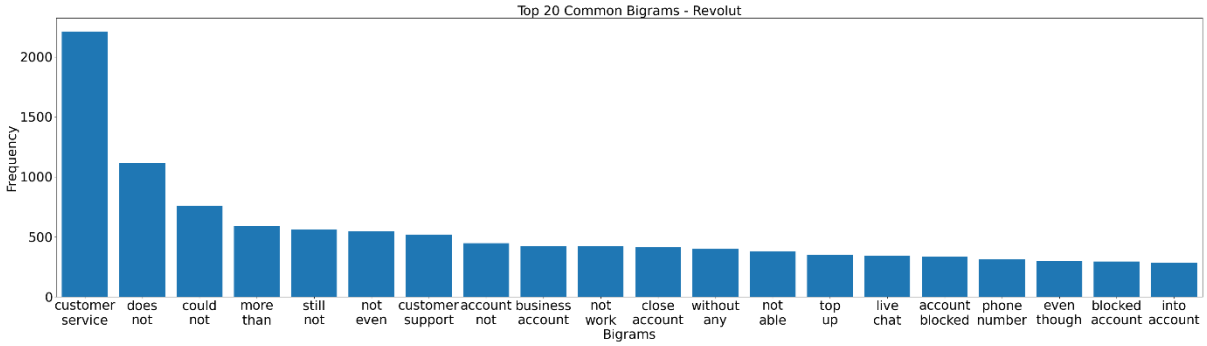

Top 20 Common Bigrams - Lloyds Bank
Frequency
0
100
200
300
400
500
customer service
could not
does not
not even
business account
more than
local branch
still not
into branch
even though
close account
open account
account not
not care
customer services
current account
waste time
not give
worst ever
not work
Bigrams

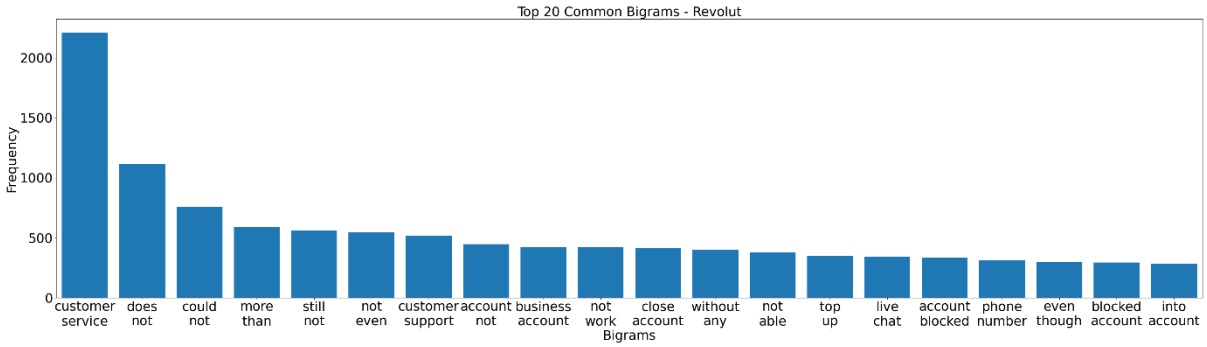

Top 20 Common Bigrams - HSBC UK
Frequency
0
200
400
600
800
1000
1200
customer service
could not
business account
does not
not even
still not
call centre
call back
customer services
local branch
current account
open account
cut off
worst ever
even though
more than
every time
into branch
not care
waste time
Bigrams

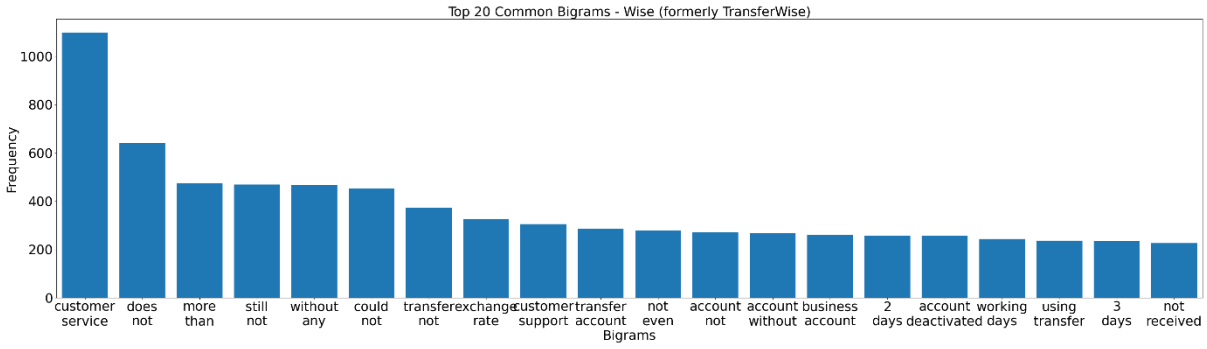

Top 20 Common Bigrams - Barclays
Frequency
0
50
100
150
200
250
300
350
customer service
could not
does not
business account
not even
cut off
speak someone
local branch
into branch
not care
still not
close account
more than
customer services
not work
open account
mobile app
even though
worst customer
worst ever
Bigrams

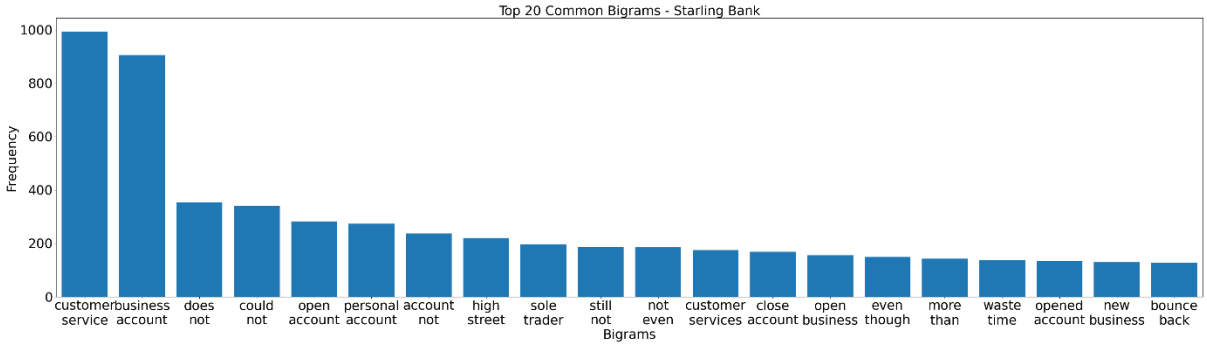

Top 20 Common Bigrams - NatWest
Frequency
0
100
200
300
400
500
600
700
customer service
could not
nat west
business account
does not
not even
fraud team
current account
still not
close account
open account
account not
into branch
closed account
call back
not care
worst ever
even though
local branch
more than
Bigrams

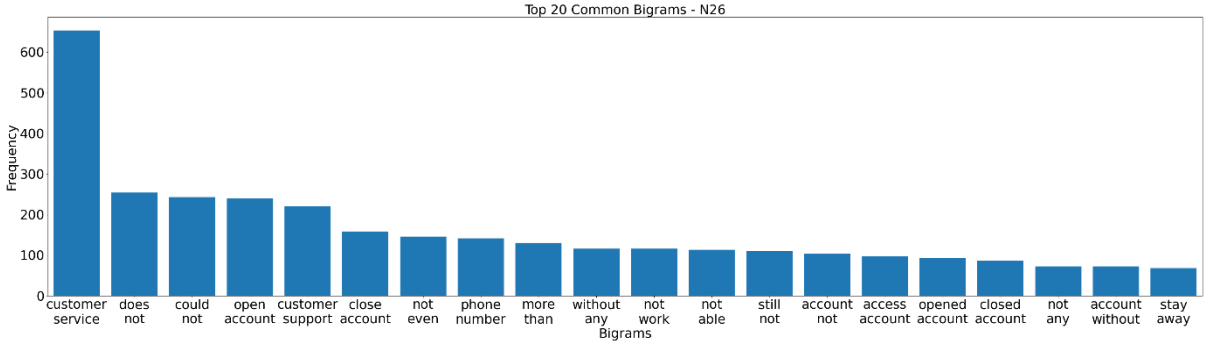

Top 20 Common Bigrams - Santander
Frequency
0
200
400
600
800
customer service
could not
does not
current account
not even
business account
even though
still not
close account
not care
more than
customer services
into branch
local branch
account not
worst ever
call back
into account
account blocked
open account
Bigrams











9.8. Appendix H – Hyperparameter Values and Results (Monzo)

Hyperparameter Combination	Hyperparameter Values			Metrics	
	UMAP		HDBSCAN		
	n_neighbors	min_dist	min_cluster	Relative Validity	Number of Clusters
1	15	0.0	15	0.292	305
2	15	0.0	30	0.222	157
3	15	0.0	60	0.161	60
4	15	0.5	15	0.163	231
5	15	0.5	30	0.126	114
6	15	0.5	60	0.021	6
7	15	0.99	15	0.576	2
8	15	0.99	30	0.045	3
9	15	0.99	60	0.001	5
10	50	0.0	15	0.230	269
11	50	0.0	30	0.249	117
12	50	0.0	60	0.211	66
13	50	0.5	15	0.139	218
14	50	0.5	30	0.026	6
15	50	0.5	60	0.088	4
16	50	0.99	15	0.156	2
17	50	0.99	30	0.005	4
18	50	0.99	60	0.004	3
19	100	0.0	15	0.210	231
20	100	0.0	30	0.214	116
21	100	0.0	60	0.189	54
22	100	0.5	15	0.521	2
23	100	0.5	30	0.007	6
24	100	0.5	60	0.005	4
25	100	0.99	15	0.080	3
26	100	0.99	30	0.004	4
27	100	0.99	60	0.004	4

9.9. Appendix I – Hyperparameter Values and Results (Revolut)

Hyperparameter Combination	Hyperparameter Values			Metrics	
	UMAP		HDBSCAN		
	n_neighbors	min_dist	min_cluster	Relative Validity	Number of Clusters

1	15	0.0	100	0.196	154
2	15	0.0	200	0.138	66
3	15	0.0	300	0.130	46
4	15	0.5	100	0.071	5
5	15	0.5	200	0.067	3
6	15	0.5	300	0.067	3
7	15	0.99	100	0.184	3
8	15	0.99	200	0.011	14
9	15	0.99	300	0.008	5
10	50	0.0	100	0.165	144
11	50	0.0	200	0.151	69
12	50	0.0	300	0.168	49
13	50	0.5	100	0.072	4
14	50	0.5	200	0.002	3
15	50	0.5	300	0.002	3
16	50	0.99	100	0.007	4
17	50	0.99	200	0.000	8
18	50	0.99	300	0.000	4
19	100	0.0	100	0.289	134
20	100	0.0	200	0.217	79
21	100	0.0	300	0.209	49
22	100	0.5	100	0.462	7
23	100	0.5	200	0.215	4
24	100	0.5	300	0.215	4
25	100	0.99	100	0.005	3
26	100	0.99	200	0.003	4
27	100	0.99	300	0.008	3

9.10. Appendix J – Cluster Topics

Cluster Number	Cluster Size	NMF Topics	LDA Topics	Assigned Topic
-1	22718	-	-	Noise
0	465	give stars, could give, one star, give one, stars because, only stars	star review, gave stars, stars because, star because, one star, give stars	Dissatisfaction
1	647	crypto currency, amount crypto, buy crypto, not buy, does not, find out	cryptocurrency exchange, premium account, buy crypto,	Crypto

			cannot send, crypto currency, cannot transfer	
2	302	trading platform, full year, buy sell, cannot buy, stock trading, gme amc	buy stocks, very very, trading account, asset value, trading platform, buy sell	Stock
3	309	hidden fees, awful company, charge back, raised charge, does not, cannot charge	without consent, cannot charge, charge fee, high fees, hidden fees, not charge	Transaction fees
4	632	stay away, away company, not recommend, recommend anyone, avoid costs, please avoid	stay clear, avoid company, not recommend, recommend anyone, stay away, avoid costs	Dissatisfaction
5	900	using app, started using, app ever, worst app, app good, service app	about app, app using, app ever, good app, downloaded app, app not	App interface
6	461	live agent, touch live, looking agent, still looking, different agents, app chat	live agent, agent not, different agents, live agent, live agent, looking agent	Live chat experience
7	310	not used, used since, used great, excellent service, not using, using again	customer service, using few, used great, more than, not used, using years	Undetermined
8	709	great service, service great, poor service, very poor, bad service, very bad	great service, bad service, good service, service ever, customer service, poor service	Dissatisfaction
9	778	good reviews, reviews here, bad reviews, good bad, negative reviews, read negative	negative reviews, not believe, positive reviews, reviews here, every day, good reviews	Dissatisfaction
10	320	account blocked, blocked since, blocked account, account took, still blocked, account still	blocked account, why blocked, just blocked, got blocked, account blocked, block account	Blocked account
11	798	metal plan, plan customer, premium account, month premium, premium plan, plan not	metal plan, premium plan, free month, premium membership, monthly fee, pay monthly	Bank subscription plans
12	648	terms conditions, read terms, does not, not work, could not, holiday could	first time, not able, could not, holiday could, does not, not work	Usage abroad
13	440	not refund, refund back, not refunded, too late, does not, refund team	not refund, does not, still waiting, chargeback team, asked refund, not receive	Refunds

14	307	app chat, replying app, customer service, service extremely, chat app, access chat	app chat, customer service, could not, friends family, not work, la app	Customer service – app experience
15	430	got scammed, just got, opening account, without any, does not, customer service	customer service, got scammed, victim fraud, could not, does not, opening account	Fraudulent transactions
16	316	bad experience, very bad, worst experience, experience ever, terrible experience, most terrible	awful experience, share experience, bad experience, experience not, experience ever, worst experience	Dissatisfaction
17	416	very disappointed, very very, not happy, really not, very frustrating, frustrating experience	very disappointed, really disappointed, not happy, extremely frustrating, very frustrating, very happy	Dissatisfaction
18	418	phone number, there phone, way contact, there way, impossible contact, contact anyone	phone number, not contact, number call, phone number, speak anyone, cannot speak	Inaccessibility to customer service
19	814	business account, open business, open account, not open, personal account, experience personal	business account, open business, business account, personal account, open account, account not	Account opening
20	672	pay bills, cannot pay, pay rent, cannot pay, verification process, through verification	could not, even though, verification process, pay hotel, cannot pay, pay bills	Online payments
21	812	still waiting, other words, still not, not resolved, solve problem, able solve	many people, solve issue, still not, not resolved, still waiting, solve problem	Unresolved ticket
22	304	goes wrong, something goes, great until, until problem, idea how, absolutely idea	went wrong, big mistake, goes wrong, something goes, not same, same mistake	Undetermined
23	345	more than, than month, next day, day same, two months, almost two	more than, next day, months ago, much time, two months, how long	Lack of response and poor turnover times
24	1045	not work, does not, top up, wanted top, never again, hard earned	again again, not rely, does not, not work, top up, first time	Undetermined
25	482	sent documents, documents requested, more documents, asking more, send documents, not find	every time, submitted documents, requested documents, same documents, send documents, more documents	Document request
26	887	absolute joke, joke company, very bad, very	joke company, very bad, not serious, really bad, far good, other than	Dissatisfaction

		very, not serious, yeah right		
27	734	good enough, not good, does not, not exist, could not, not happen	could not, not even, does not, not happen, good enough, not good	Undetermined
28	568	speak human, human being, automated response, standard automated, does not, security system	human being, speak human, cannot speak, speak anyone, compliance team, automated system	Poor responses
29	883	express delivery, free express, still not, not received, new one, order new	express delivery, customer service, new one, still not, not arrived, tracking number	Card delivery fees
30	491	live chat, chat not, via chat, chat team, online chat, every time	live chat, chat not, contacted chat, chat history, online chat, every time	Live chat experience
31	339	business elsewhere, take business, bad business, business practices, run business, business cannot	run business, business model, bad business, not even, business elsewhere, take business	Business accounts
32	711	phone number, changed phone, verify identity, cannot verify, does not, not allow	phone number, drivers license, driving licence, does not, driving license, verify account	Account verification
33	743	customer service, service response, app chat, via app, chat app, live chat	app chat, app not, chat app, live chat, customer service, there way	Customer service – app experience
34	2483	copy paste, paste reply, reply below, update reply, more than, than month	not answer, not even, still not, live chat, copy paste, more than	Poor responses
35	3739	customer service, worst customer, customer support, contacted customer, support team, chat support	customer support, support chat, customer service, customer services, customer service, customer support	Customer service – non-app experience
36	1209	transaction not, not through, could not, not even, customer service, contacted customer	could not, still not, transaction not, does not, transactions not, fraudulent transaction	Fraudulent transactions
37	657	made payment, 10th march, payment processor, contact payment, payment not, not through	payment processor, payment not, weeks ago, payments not, into account, made payment	Payment issues
38	517	apple pay, via apple, does not, not work, top up, used apple	could not, apple pay, mastercard atm, atm app, not working, google pay	Payment issues

39	988	customer service, up account, customer support, blocked account, account locked, locked over	blocked account, business account, customer service, locked out, account blocked, customer support	Blocked account
40	334	tax return, asked tax, tax returns, company tax, asked provide, invoices etc	months statements, last three, tax return, tax returns, asked provide, last months	Invoice request
41	1071	source funds, verify source, access funds, not access, source income, verify source	source income, could not, source funds, verify source, access funds, proof funds	Frozen account
42	574	close account, account not, closing account, account down, closed account, account after	closing account, closed account, close account, account not, down account, delete account	Account closure
43	690	blocked account, account reason, account blocked, without any, block account, account reason	not able, account closed, account blocked, access account, blocked account, account without	Blocked account
44	545	exchange rates, good exchange, exchange rate, best exchange, currency exchange, like currency	currency exchange, foreign currency, multiple currencies, more than, exchange rates, exchange rate	Exchange rates
45	1150	not transfer, could not, cannot transfer, transfer funds, transfer account, tried transfer	made transfer, transfer not, into account, cannot transfer, not transfer, still not	Transfers
46	372	atm withdrawal, free atm, cash withdrawal, charged cash, withdraw cash, cash atm	atm withdrawal, cash withdrawal, atm machine, cash withdrawals, withdraw cash, take out	ATM withdrawal
47	463	euro account, account euro, uk account, transfer uk, gbp account, even though	euro account, uk account, gbp account, into account, eur gbp, top up	Cross currency transfers

9.11. Appendix K – Regrouped Cluster Topics

New Cluster Number	Cluster Size	Old Regrouped Clusters	Assigned Topic
-1	29314	-1, 0, 4, 7, 8, 9, 16, 17, 22, 24, 26, 27	Noise
0	647	1	Crypto
1	302	2	Stocks
2	900	5	App interface
3	798	11	Bank subscription plans
4	648	12	Usage abroad
5	440	13	Refunds

6	418	18	Inaccessibility to customer service
7	814	19	Account opening
8	812	21	Unresolved ticket
9	816	25, 40	Tedious paperwork (request of documents)
10	1192	3, 29	Additional fees (e.g. card delivery and transaction fees)
11	1520	6, 28, 30	Live chat experience
12	339	31	Business accounts
13	711	32	Account verification
14	1050	14, 33	Customer service – app experience
15	2828	23, 34	Lack of response and poor turnover times
16	3739	35	Customer service – non-app experience
17	1639	15, 36	Fraudulent transactions
18	1846	20, 37, 38	Online payments
19	1998	10, 39, 43	Blocked account
20	1071	41	Frozen account and money holdout
21	574	42	Account closure
22	545	44	Exchange rates
23	1150	45	Transfers
24	372	46	ATM withdrawal
25	463	47	Cross currency transfers

9.12. Appendix L – Binary Random Forest Model Performance (All Clusters)

Cluster Number	Dataset	Label	Metrics			
			Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
-1	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	69	68	72	70
		1	69	70	66	68
0	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	98	98	98	98
		1	98	98	98	98
1	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	94	98	96
		1	96	97	94	96

2	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	95	97	96
		1	96	97	95	96
3	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	94	98	96
		1	96	98	94	96
4	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	93	93	93
		1	93	93	93	93
5	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	92	98	95
		1	95	98	91	94
6	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	91	96	93
		1	93	95	91	93
7	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	89	86	92	89
		1	89	92	85	88
8	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	94	95	95
		1	95	95	94	95
9	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	95	96	96
		1	96	96	95	96
10	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	88	85	90	88

		1	88	90	85	87
11	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	87	86	89	87
		1	87	89	86	87
12	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	92	90	95	92
		1	92	95	90	92
13	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	93	96	95
		1	95	96	93	95
14	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	85	81	90	85
		1	85	89	79	84
15	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	93	92	93
		1	93	92	93	93
16	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	92	94	93
		1	93	94	93	94
17	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	90	88	93	90
		1	90	93	88	90
18	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	88	86	91	88
		1	88	90	86	88
19	Training	0	100	100	100	100
		1	100	100	100	100

	Validation	0	92	90	93	92
		1	92	93	90	92
20	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	94	96	95
		1	95	96	94	95
21	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	95	97	96
		1	96	97	95	96
22	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	97	96	98	97
		1	97	98	95	97
23	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	97	96	98	97
		1	97	98	96	97
24	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	94	99	96
		1	96	99	94	96
25	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	96	94	95
		1	95	94	96	95

9.13. Appendix M – Final Regrouped Cluster Topics

Final Cluster Number	Cluster Size	Old Regrouped Clusters	Assigned Topic
-1	29314	-1, 0, 4, 7, 8, 9, 16, 17, 22, 24, 26, 27	Noise
0	647	1	Crypto
1	302	2	Stocks
2	900	5	App interface

3	798	11	Bank subscription plans
4	648	12	Usage abroad
5	440	13	Refunds
6	10367	14, 21, 30, 33, 34, 35,	Customer service
7	3386	10, 19, 39, 42, 43	Account issues
8	816	25, 40	Tedious paperwork (request of documents)
9	883	3, 29	Additional fees (e.g. card delivery and transaction fees)
10	648	31	Business accounts
11	711	32	Account verification
12	1639	15, 36	Fraudulent transactions
13	1846	20, 37, 38	Online payments
14	1071	41	Frozen account and money holdout
15	545	44	Exchange rates
16	1613	45, 47	Transfers
17	372	46	ATM withdrawal

9.14. Appendix N – Binary Random Forest Model Results (All Regrouped Clusters)

Cluster Number	Dataset	Label	Metrics			
			Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
-1	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	70	69	74	71
		1	70	72	66	69
0	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	97	97	98	97
		1	97	98	97	97
1	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	94	96	95
		1	95	96	94	95

2	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	94	97	96
		1	96	97	94	96
3	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	94	98	96
		1	96	98	94	96
4	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	92	94	93
		1	93	94	91	93
5	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	94	91	98	94
		1	94	98	90	94
6	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	93	91	92	92
		1	93	94	93	93
7	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	91	88	93	91
		1	91	94	88	91
8	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	95	98	96
		1	96	98	95	96
9	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	87	82	95	88
		1	87	94	80	86
10	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	90	92	88	90

		1	90	88	93+	90
11	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	93	97	95
		1	95	97	93	95
12	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	89	88	91	90
		1	89	91	87	87
13	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	87	84	90	87
		1	87	90	84	87
14	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	94	93	95	94
		1	94	95	93	94
15	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	96	97	96
		1	96	97	96	96
16	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	95	94	97	95
		1	95	97	94	95
17	Training	0	100	100	100	100
		1	100	100	100	100
	Validation	0	96	95	97	96
		1	96	97	95	96

9.15. Appendix O – Link to Datasets, Notebooks, and Results

The datasets, notebooks used for each component within the pipeline, and their corresponding results are organised and accessible via the (hyper)link provided.

9.16. Appendix P - Project Proposal

Natural Language Processing in Tracking a Company's Performance

1. Introduction

As any industry matures, key market players are identified and cemented as giants within it. To capture the limited customer pool within the industry, tight competition arises with continuous innovation of new products and services, and improvements onto existing ones, often serving as key ingredients and centrepieces to not only customer acquisition but customer retention as well. Thus, it is in a company's best interest to understand customer sentiment and needs, and implement proactive strategies (e.g. improving customer service, customer experience or user-interface of websites) rather than reactive ones. Even in large industries like UK's banking sector do we see similar narratives with (initially) smaller companies standing with giants due to innovation and entrepreneurship drawn from customer insights. A sector dominated by traditional banks like Barclays and HSBC, experienced tremors and disruption with the introduction of digital challenger banks like Wise and Starling Bank in 2011 and 2014 respectively – an introduction that was successful largely due to identifying gaps, and subsequently opportunities, in currency conversion and overseas transfers respectively.

Though inferences on customer needs can be drawn from internal data (e.g. spending behaviour, time spent on webpages or frequency of visit), customer reviews on external platforms, like Trustpilot or Facebook, may be a lower hanging fruit despite the arguably more unstructured format of data. And with the immense volume and continuous stream of customer reviews and feedback, it is not unlikely for the latter to hold true. While reviews of the company itself provides a good indication of what customers like and dislike about its offerings and services, competitor reviews greatly supplements it as a learning exercise to identify good practices which can be adopted and gaps in their offerings and services to be capitalised upon.

This project will be completed in collaboration with and supported by Kainos Group plc, whereby the purpose of this project is to determine whether insights on potential improvements to a company's offerings and services can be inferred from data scraped through consumer review websites - Trustpilot in particular. Using Natural Language Processing (NLP) techniques like sentence tokenisation, word embedding and topic modelling, we will look to build topic classification models which identifies the topic of discussion for each sentence (and review). As a case study, we'll look at English reviews left on Trustpilot of banks with presence within the UK - both traditional and digital challenger ones.

In this project we will look to employ an NLP pipeline which leverages on unsupervised learning techniques prior to migrating over to supervised learning methods. In addition to word embeddings and clustering of sentences, topic modelling techniques will be adopted to draw initial insights on topics amongst the reviews which will be utilised to build a suite of topic classification models. Additionally, both, unsupervised and supervised learning methods will be evaluated through internal (Thalamuthu *et*

al. 2006) and external (Dudoit and Fridlyand, 2002) measures respectively. Despite that, expert judgement and human reasoning will supplement statistical measures and remain as the cornerstone to decision making.

The objectives of this project are as follows:

- To scrape customer reviews, supporting information (e.g. customer information) and, metadata (e.g. time and location of review), of UK banks from TrustPilots' webpages
- To consider, iteratively apply and discuss different word embedding models and techniques onto the processed corpus of customer reviews
- To leverage on the clusters and covariates (i.e. sentiment of review, customer information and metadata) and develop topic classification models
- To compare and contrast the performance and quality of different word embedding models and clustering techniques through internal measures
- To assess the performance of topic classification models through external measures

The research question is formulated as follows:

Can word embeddings and topic modelling on customer reviews derive insights on potential improvements to a company's offering and services?

The main and apparent beneficiaries to this project are retail companies; getting an understanding of how one's offerings and services are performing relative to its competitors allow for efficient prioritisation and gaining a marginal edge. With non-retail companies, the customer base are other companies and hence reviews may be sparse. However, methods and insights that are to be applied and drawn upon are transferable onto other research projects.

2. Critical Context

2.1. Overview

As individuals continue to increase their online presence through online platforms, a large influx of data is suddenly at our disposal. Each platform serves its own niche within the market, and TrustPilot is no exception. As an online platform, TrustPilot serves as a consumer review website allowing users to leave reviews of companies based on their experience and summarised by a star-rating from one to five - worst and best respectively.

Though popular approaches like Term Frequency (TF) and Term Frequency - Inverse Document Frequency (TF-IDF) provide a convenient way of understanding the importance of words relative to a corpus (Rajaraman and Ullman, 2011), it utilises a Bag-Of-Words (BOW) approach which may fail to retain semantic meaning of the review. Contrastingly, word embedding techniques aim to capture

semantic meaning of words under the assumption that words which share the same context are semantically similar (Mandelbaum and Shalev, 2016). Ultimately, word embeddings will position reviews spatially whereby reviews with similar semantic meaning and topics are spatially closer together.

With a real-world classification task, one main challenge is that the data is often unlabelled and so granular insights may be difficult to produce (i.e. an unsupervised learning task). However, clustering serves as a way to identify any underlying patterns and/or structure within our dataset in a non-parametric way (i.e. no prior beliefs or assumptions) (Alashwal *et al.* 2019). For this project, it will help to group reviews with similar semantics and topics together.

2.2. Data Pre-Processing

Khurana *et al.* (2022) summarised and presented the state of the art within the NLP field. Aside from showcasing a current snapshot of the NLP field and discussing future trajectories, they identify key elements in understanding text, or the practice of Natural Language Understanding (NLU). Within it, they highlight the importance of various aspects of linguistics in addition to semantics – discourse and pragmatics. Discourse surrounds inferred meaning through a logical structure of more than one sentence while pragmatics captures it through context. Without either, ambiguity arises in understanding a piece of text. Additionally, they recap common text-cleaning processes like tokenisation, stop word removal, stemming and lemmatisation but caveat it with the risk of altering the semantic meaning of a sentence.

A potential challenge with real-world data is that it can be sparse, especially when the domain is very specialised or niche. Edunov *et al.* (2018) explains the idea of back-translation whereby synthetic data points are generated through translating a sample of text to a different language and back to the original language. The synthetic translations are then added and used for training. They proposed the idea of the sampling and adding noise to the conventional back-translation outputs and found model performance to be higher compared to without it.

2.3. Word Embedding and Topic Modelling

To understand the semantics of text-based data, word embedding and topic modelling techniques have been developed and built upon. Landauer, Foltz and Laham (1998) introduces and writes about the technicalities surrounding Latent Semantic Analysis (LSA). They explain how LSA does not solely rely on co-occurrences of words within the text, nor manually constructed dictionaries, but instead applies singular value decomposition (SVD) onto a word-document occurrence matrix to decompose it into a series of three components: a word-topic matrix, topic-document matrix and a diagonal matrix of scalars or a representation of the topic strength. Similarity measures can then be derived from the product of these components.

Mikolov *et al.* (2013) proposed two new model architectures for word representation which is commonly coined today as "word2vec" – a state-of-the-art word embedding model. They reiterate the benefits of neural networks in the preservation of linear regularities among words. They introduce a Continuous Bag-of-Words (CBOW) and Continuous Skip-gram (Skip-gram) model as alternatives and improvements to vector representations of words. Both utilise a feedforward neural net language model (NNLM) but CBOW predicts the word based on its context (i.e. surrounding words) while Skip-gram does the opposite and predicts the surrounding words based on a single word. From it, they found improvements in accuracy and computational intensity as compared to previously renowned techniques.

As an improvement to both, Pennington, Socher and Manning (2014) instead utilised Global Vectors for word representation, or commonly referred to as "GloVe" – another state-of-the-art approach. They highlight how LSA has found poor results in word analogy tasks while word2vec works with only a series of small local context of words. Instead, GloVe works on a word-word co-occurrence matrix over the entire corpus and displays it in a probabilistic manner. From word analogy and similarity tasks, over different hyperparameters, they found GloVe to outperform its counterparts like SVD and CBOW.

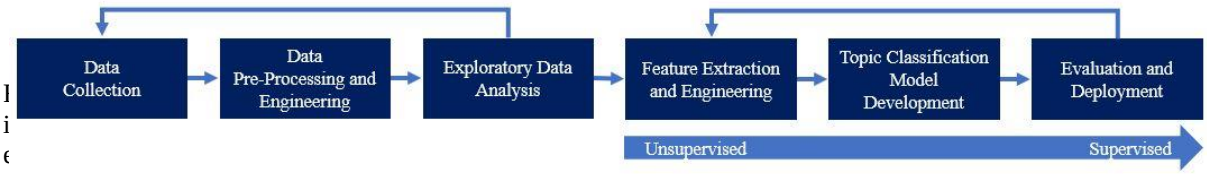
2.4. Clustering

Campello *et al.* (2013) generalises clustering algorithms to fall under a set of orthogonal features: hierarchical and flat and centroid and density based. However, they identify a set of limitations with the current approaches (e.g. flat clustering algorithms may lead to erroneous clustering given the single density threshold and existing hierarchical approaches fail to allow for easy interpretation of most significant clusters). Instead, they introduce a new clustering algorithm, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which clusters a set of datapoints based on the mutual reachability distance between each pair of points – a metric which considers the distance between them and the distance to the kth nearest neighbour. Applying it a set of popular datasets like "Iris" and "Wine", HDBSCAN to outperform its counterparts for a majority of the popular datasets.

3. Approaches

3.1. Overview

This project looks to implement a semi-supervised learning pipeline whereby unsupervised learning methods are initially employed prior to moving over to supervised learning. The general workflow of this project is laid out in Figure 1.



I
i
e

3.2. Data Collection

Customer reviews posted on TrustPilot will make up this project's dataset whereby it'll be scraped using BeautifulSoup, Selenium and/or srapy - open-source libraries for web-crawling in Python. With it,

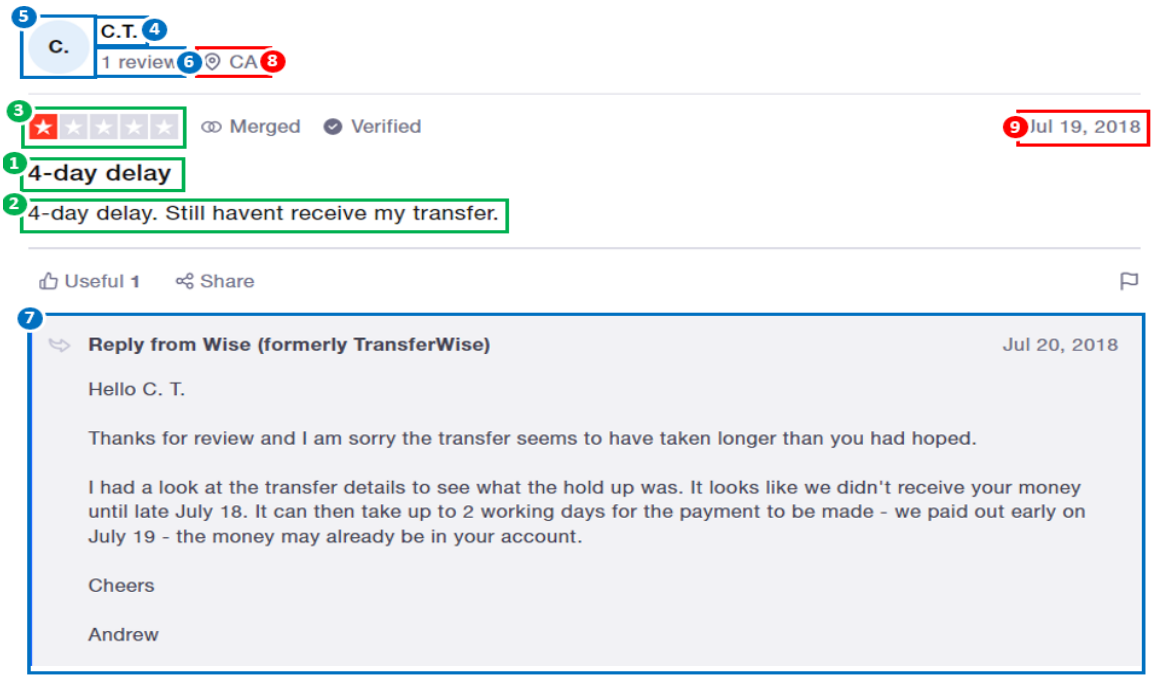


Figure 2: Relevant Information in a Review. 1: Title, 2: Review text, 3: Rating given, 4: User's name, 5: User's profile picture, 6: Number of reviews left by the user, 7: Company reply, 8: Location, 9: Date of review

relevant information can be scraped and obtained as laid out in Figure 2.

The elements marked in green, blue and red capture the three main types of information that'll be scraped, namely: the review itself, supporting information and metadata respectively.

3.3. Data Pre-Processing and Engineering

The scraped data will be returned as a combination of Comma-separated Values (csv) and JavaScript Object Notation (json) objects, depending on the size and complexity of the dataset – csv files allow for easier ad-hoc visualisation but runs into limitations (e.g. number of characters stored per cell). The main information components as discussed in Section 3.2 will be stored and subsequently unpacked into dataframes. Thereafter, the text of reviews will be cleaned using approaches adapted from Sriram *et al.* (2010) and Khurana *et al.* (2022) to ultimately improve the quality of the extracted features further down the workflow:

1. Sentence tokenisation – splitting a review or piece of text into individual sentences
2. Stop-word removal – removal of common words with little semantic meaning
3. Punctuation and character removal – removal of certain punctuation and characters

Additionally, other pre-processing steps will be taken, such as:

1. Duplicated review and sentence removal – removal of repeated reviews and/or tokenised sentences
2. Reviews' text proxy – use a reviews' title as a proxy to its text if there is no text
3. Missing value removal – removal of any entries without a tokenised sentence
4. Outlier removal – removal of outliers and non-sensical sentences

Potentially having multiple datasets which are associated but not identical, data engineering will be required to manage and string them together (e.g. left-join, concatenation, merging).

3.4. Exploratory Data Analysis

Prior to proceeding to feature extraction and engineering, the size, extensiveness, and context of the resulting dataset will need to be evaluated. Frequency distributions of review lengths, unigrams and bigrams used, and user demographic (e.g. location) will provide a good indication to the quality of the abovementioned features. Additionally, moving average, marginal and cumulative time series plots of review frequencies, rating and reply rates will further supplement this. If required, data collection and/or data pre-processing can be revisited to further ensure that the dataset is comprehensive and processed appropriately.

3.5. Feature Extraction and Engineering

Feature extraction and engineering will encapsulate five main components sequentially: word embedding, dimensionality reduction, clustering, topic modelling and evaluation. Generally, word embeddings bring our text-based data to the numerical space in accordance to semantic meaning (e.g. words "similar" and "alike" will be closer together versus "similar" and "different") and allow for further processing. Dimensionality reduction then reduces the word embeddings down to fewer dimensions, allowing for visualisation and clustering of reviews and sentences into groups in accordance to similarity (e.g. semantic similarity). Topic modelling supplies time savings by identifying topics for each cluster without human-tagging. The quality of embeddings, clustering and topics identified will then be evaluated with a combination of metrics and human judgement. Since the abovementioned components are costly, the methodology laid out from Sections 3.5 to 3.7 will first be performed over a sub-sample of our dataset (e.g. for a single bank) prior to deploying it to the entire dataset.

Our tokenised sentences will be translated onto word embeddings using universal sentence encoders – two encoding models built upon transformers and a deep averaging network (DAN) which was found to outperform word2vec and GloVe (Cer *et al.*, 2018). Both encoding models output a 512-dimensional sentence embedding and considers the context of the word, but the transformer-based encoding model computes the element-wise sum at every word position while the DAN-based encoding model averages it out prior to parsing it through the next layer in the deep neural network (DNN). They

found the transformer-based model to perform best but was caveated by its computational intensity. Nonetheless, we'll look to implement and compare them both via quality of embeddings and computation time.

To visualise our sentence embedding, we will look to reduce its dimensionality down to either 3-dimensions (3D) or 5-dimensions, allowing for a singular 3D or 10 triad-wise 3D plots respectively. We will look to implement a state-of-the-art dimensionality reduction technique, namely Uniform Manifold Approximation and Projection (UMAP), which performs better than popular techniques like Principal Component Analysis (PCA) and rivals the prior best approach: t-distributed Stochastic Neighbour Embedding (t-SNE) (McInnes, Healy, and Melville, 2020). UMAP preserves more of the data's global structure in tandem with the local structure versus its counterparts, along with lower computational intensity as well.

As discussed in Section 2.4, HDBSCAN serves as a hierarchical-density-based clustering approach (i.e. a non-parametric approach that allows for relationships between clusters in a hierarchical manner) which clusters our sentence embeddings based on the mutual reachability distance between pairs of points and given that the embeddings are dispersed in accordance to semantics, clusters will be grouped with respect to semantics as well.

With the newly formed clusters, we aim to gain an understanding of the clusters by identifying a set of common topics which govern each cluster. For this, we will look to implement Latent Dirichlet Allocation (LDA) which essentially outputs a pre-determined number of topics by evaluating the probabilities of words belonging to a document and words belonging to a topic. Ultimately these topic probabilities serve as a representation of each document (each sentence and cluster in this case) (Blei, Ng, and Jordan, 2003).

Given that the quality of our dimensionality reduction and clustering (and hence, generated topics as well) are contingent on a set of hyperparameters (e.g. our preference of preserving the local vs global structure for UMAP and the minimum number of points to constitute a cluster for HDBSCAN), we will perform hyperparameter tuning and select the best performing one qualitatively and quantitatively. Qualitatively, we will evaluate the quality of the clustering via the number of clusters and spatial positions in the 3D space. Ideally, there would be a fair number of clusters which are well separated. Additionally, the generated topics will be considered to ensure that topics are sensical. Quantitatively, given it is an unsupervised learning approach, we will utilise internal measures (Thalamuthu *et al.* 2006) like Density-Based Clustering Validation (Moulavi *et al.,* 2014)) to compare hyperparameter combination.

3.6. Topic Classification Model Development

Prior to model development, the sub-sample of datapoints will be split into training and testing using a 90-10 split respectively. Contingent on the distribution of clusters and cluster sizes, significantly larger clusters may be down-sampled in the training set to avoid biased predictions towards a certain cluster (e.g. if 99% of the dataset is made up of points from cluster 1 then predicting the entire remaining 1% wrongly still results in an accuracy of 99%). Additionally, when developing the classification models, we will implement k-fold cross validation whereby the training set is divided into k number of folds and each fold will be used as holdout or validation while the remaining k-1 folds will be used for training. The validation sets and its respective predictions will be stored and aggregated through majority voting.

We look to build topic classification models, Random Forests specifically, to classify a sentence as a given cluster, and hence topic. Random Forests are a combination of decision trees which mitigates some of the overfitting risk that stems from individual decision trees by not only determining the number of trees to make up the random forest, but the maximum depth of each tree and the maximum number of features to consider at each split (Breiman, 2001).

Aside from multiclass classification (i.e. a sentence is predicted to be either in cluster 1, 2, or 3) we will look to implement a series of one-vs-rest (OvR) classification models as well which act as binary classifiers (i.e. a sentence is predicted to be in cluster 1 or not in cluster 1). A binary classifier will be developed for each cluster and all binary classifiers will be applied onto each sentence to assess for any multi-topic-tagging. Grid search will be utilised to tune the hyperparameters for each random forest with accuracy serving as the decision metric.

3.7. Evaluation and Deployment

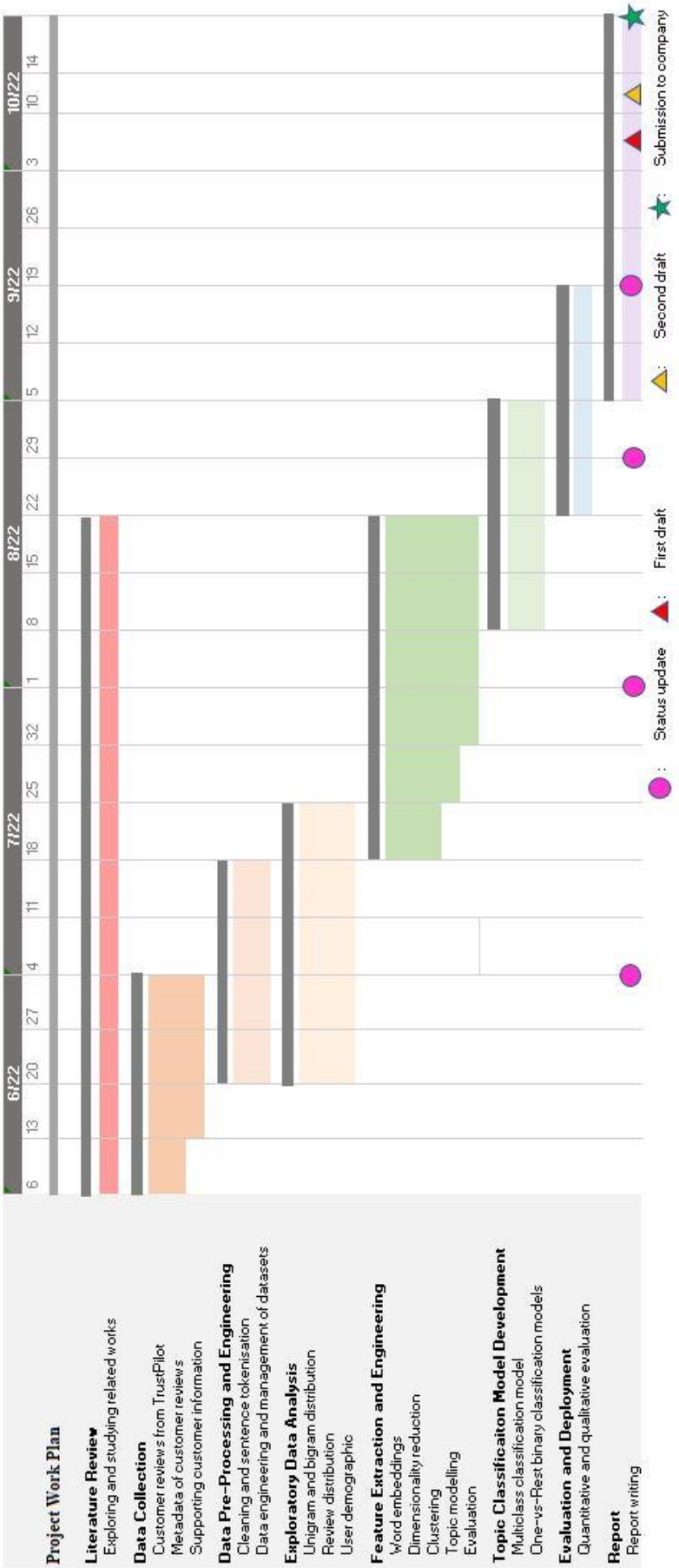
To evaluate the trained models, external quantitative metrics like accuracy, recall, and f-score will be measured and evaluated. Additionally, confusion matrices serve as a convenient way to evaluate the number of false positives and false negatives while the probability distribution of classification probabilities for the true positives, true negatives, false positives and false negatives will act as a way to evaluate the performance of our models. Ideally, there would we be no false positives or false negatives and only true negatives and true positives and the probability distribution being a multimodal distribution – two peaks: one at probability of 0 (i.e. 100% certain that a sentence does not fall within a cluster) and another at 1 (i.e. 100% certain that a sentence falls within a cluster); though it may be a case of overfitting instead. Overfitting can be further assessed by comparing the errors in the training and validation set. In the presence of it, validation errors will be high while training errors are low.

With the presence of false negatives and false positives, we would hope for it to follow a similar distribution to the ideal case mentioned above. For example, consider a false negative whereby a sentence that falls within an account opening cluster is misclassified as a sentence about customer service instead with a significantly high probability. Then human judgement allows for us to evaluate

whether the false negative is warranted and whether reassignment of sentences to different clusters is needed. Doing so allows for evaluation of the quality of clustering and performance of the model. To formalise this process, we will set thresholds on the probability of classification whereby false positives and false negatives with classification probabilities that exceed the threshold will be reassigned to the predicted class and the class which makes up the majority of the false negatives respectively.

Finally, the final model or set of models will be deployed and applied onto the testing set and the remainder of the dataset (i.e. other banks). Thereafter analysis will be performed to identify any trends and further insights.

4. Work Plan


Project Work Plan
Literature Review
Exploring and studying related works
Data Collection
Customer reviews from TrustPilot
Metadata of customer reviews
Supporting customer information
Data Pre-Processing and Engineering
Cleaning and sentence tokenisation
Data engineering and management of datasets
Exploratory Data Analysis
Unigram and bigram distribution
Review distribution
User demographic
Feature Extraction and Engineering
Word embeddings
Dimensionality reduction
Clustering
Topic modelling
Evaluation
Topic Classificaiton Model Development
Multiclass classification model
One-vs-Rest binary classification models
Evaluation and Deployment
Quantitative and qualitative evaluation
Report
Report writing
6/22
13
20
27
4
11
18
25
32
1
8
15
22
29
5
12
19
26
3
10
14
7/22
8/22
9/22
10/22
Status update
First draft
Second draft
Submission to company

Whilst scraping data, a subsample can first be extracted as a proof-of-concept (PoC) to test any hypothesis and code informally. Large overlap between data collection, data pre-processing and engineering, and EDA is due to the recursive process. The same is for feature extraction and engineering, topic classification model development and evaluation and deployment.

5. Risks

Risk	Likelihood (1-5)	Consequence (1-5)	Impact (L x C)	Mitigation Plan
Blocking of data scraping bots on TrustPilot	5	2	10	Implement delays between pull requests, scrape posts in batches and utilise virtual private networks (VPN)
Bottlenecks from data collection	4	3	12	Sample smaller batch of reviews as a PoC to test code and hypothesis' while performing larger jobs
Non-extensive dataset	2	4	8	Source reviews from other websites (e.g. Facebook) or consider other industries
Computational intensity of feature extraction and model development	5	2	10	Leverage on multiprocessing and virtual machines to speed up the computational time and reduce the computational intensity on local machines
Lack of technical knowledge	2	5	10	Continuous learning from literature review and seek advice from project supervisor
Unfamiliarity with new Python libraries	3	3	9	Refer to examples and official documentation
Falling behind on schedule	1	5	5	Constantly refer to work plan and periodic status updates with university supervisor
Loss of code/data from override or crashes	3	4	12	Version control using GitHub and storing files virtually

6. References

Alashwal, H. *et al.* (2019) 'The Application of Unsupervised Clustering Methods to Alzheimer's Disease', *Frontiers in Computational Neuroscience*, 13. Available at: https://www.frontiersin.org/articles/10.3389/fncom.2019.00031

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) 'Latent dirichlet allocation', *The Journal of Machine Learning Research*, 3(null), pp. 993–1022.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324.

Campello, R.J.G.B., Moulavi, D. and Sander, J. (2013) 'Density-Based Clustering Based on Hierarchical Density Estimates', in J. Pei et al. (eds) *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 160–172. Available at: https://doi.org/10.1007/978-3-642-37456-2_14.

Cer, D. *et al.* (2018) 'Universal Sentence Encoder'. arXiv. Available at: http://arxiv.org/abs/1803.11175

Dudoit, S. and Fridlyand, J. (2002) 'A prediction-based resampling method for estimating the number of clusters in a dataset', *Genome Biology*, 3(7), p. research0036.1. Available at: https://doi.org/10.1186/gb-2002-3-7-research0036.

Edunov, S. *et al.* (2018) 'Understanding Back-Translation at Scale', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. *EMNLP 2018*, Brussels, Belgium: Association for Computational Linguistics, pp. 489–500. Available at: https://doi.org/10.18653/v1/D18-1045.

Khurana, D. *et al.* (2022) 'Natural language processing: state of the art, current trends and challenges', *Multimedia Tools and Applications* [Preprint]. Available at: https://doi.org/10.1007/s11042-022-13428-4.

Mandelbaum, A. and Shalev, A. (2016) 'Word Embeddings and Their Use In Sentence Classification Tasks'. arXiv. Available at: http://arxiv.org/abs/1610.08229 (Accessed: 16 July 2022).

McInnes, L., Healy, J. and Melville, J. (2020) 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. arXiv. Available at: https://doi.org/10.48550/arXiv.1802.03426.

Moulavi, D. *et al.* (2014) 'Density-Based Clustering Validation', in *Proceedings of the 2014 SIAM International Conference on Data Mining*. *Proceedings of the 2014 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, pp. 839–847. Available at: https://doi.org/10.1137/1.9781611973440.96.

Pang, B. and Lee, L. (no date) 'Opinion mining and sentiment analysis', p. 94.

Rajaraman, A. and Ullman, J.D. (eds) (2011) ‘Data Mining’, in *Mining of Massive Datasets*. Cambridge: Cambridge University Press, pp. 1–17. Available at: https://doi.org/10.1017/CBO9781139058452.002.

Thalamuthu, A. *et al.* (2006) ‘Evaluation and comparison of gene clustering methods in microarray analysis’, *Bioinformatics*, 22(19), pp. 2405–2412. Available at: https://doi.org/10.1093/bioinformatics/btl406.