APP REVIEWS ANALYSIS USING DEEP LEARNING ALGORITHMS: A CASE STUDY OF MONZO APP

BY

SUBMITTED TO

FACULTY OF BUSINESS, LAW, AND DIGITAL TECHNOLOGIES
SOUTHAMPTON SOLENT UNIVERSITY

Supervisor:

January 20, 2023

A dissertation submitted in partial fulfilment of the requirements of Solent University for the degree of MSC Applied AI & Data Science

o

Acknowledgement

Abstract

The use of Mobile applications has gained more traction in recent times due to advancing technologies and their personalised ease of use across different sectors. This is owed largely to the fact that mobile applications have become not just a tool for messaging and communications, but rather a one-stop place where humans can perform a varying number of activities. In recent times, the finance apps category has been gaining more traction as their apps allow users to execute personalized financial operations from the comfort of their phones. In the same vein, reviews from users of these applications posted across app marketplaces now serve as a very rich resource for app developers to mine insights on how to quickly evolve and improve features of their apps to remain relevant in a competitive app world eco-system. In mining insights from apps review, several techniques have been utilised in the past with underlying limitations however, in recent times, machine learning technologies have seen improved use due to their promising capabilities in text mining tasks. In this research,
This study utilizes natural language processing to do sentiment analysis and topic modelling on the reviews from the Monzo application , one of the top 10 online banking apps in the UK as of as of today.With a dataset of 12,863 records of the apps review dataset, gotten from the google play store, the LSTM, BERT, LDA, LR, SVM, and MNB, models are utilised for both sentiment analysis and topic modelling on the app's reviews. For the sentiment analysis task, the LSTM model performed best of all models with an accuracy score of 90.4% and an F1 score of 94%. By comparing the performance of the BERT and LDA models for topic modelling, the BERT model was better performing in identifying 168 distinct topics contained in the reviews, providing further insights on areas of concentration for future development.

Table of Contents

List of Tables

List of Figures

ACRONYMS

AI -Artificial Intelligence
BERT -Bidirectional Encoder Representations from Transformers
CNN -Convolutional Neural Network
CSV -Comma-Separated Values
DL- Deep learning
GPT- Generative Pre-trained Transformer
Ios- iPhone Operating System
JSON- JavaScript Object Notation
LDA- Latent Dirichlet Allocation
LR- Logistic Regression
LSTM -Long Short-Term Memory
ML- Machine Learning
MNB- Multinomial Naïve Bayes
NEO- New
NLP -Natural Language Processing
NLTK- Natural Language Toolkit
NN- Neural Network
PDF- Portable Document Format
RNN- Recurrent Neural Network
SVM -Support Vector Machine
TF-IDF- Term Frequency-Inverse Document Frequency
US- United State
UK- United Kingdom

1.0 Introduction

The rate at which mobile applications are being developed is on the rise every day with each promising user with features that can help them perform tasks faster and better. This is also because mobile gadgets have become not just a tool for messaging and communications, but rather a one-stop place where humans can perform a varying number of activities. Taylor et al, (2011) defined mobile apps as "small programs that run on a mobile device and perform tasks ranging from banking to gaming and web browsing". These apps provide functionalities in fields across education, payments, gaming, health and fitness, eCommerce, and transportation to mention a few.

In recent times, mobile apps have seen increased use over desktop apps, as the former is more user-friendly, customizable, less expensive, and more accessible to download, Roy et al. (2011). There are roughly 3.5 million applications in various categories on the Google Play store, 1.6 million apps on the iOS store, according to a statista report published in October 2022. These apps have a market value of 187.58 billion, and through 2030, there is expected to be a 13.4% annual growth rate.

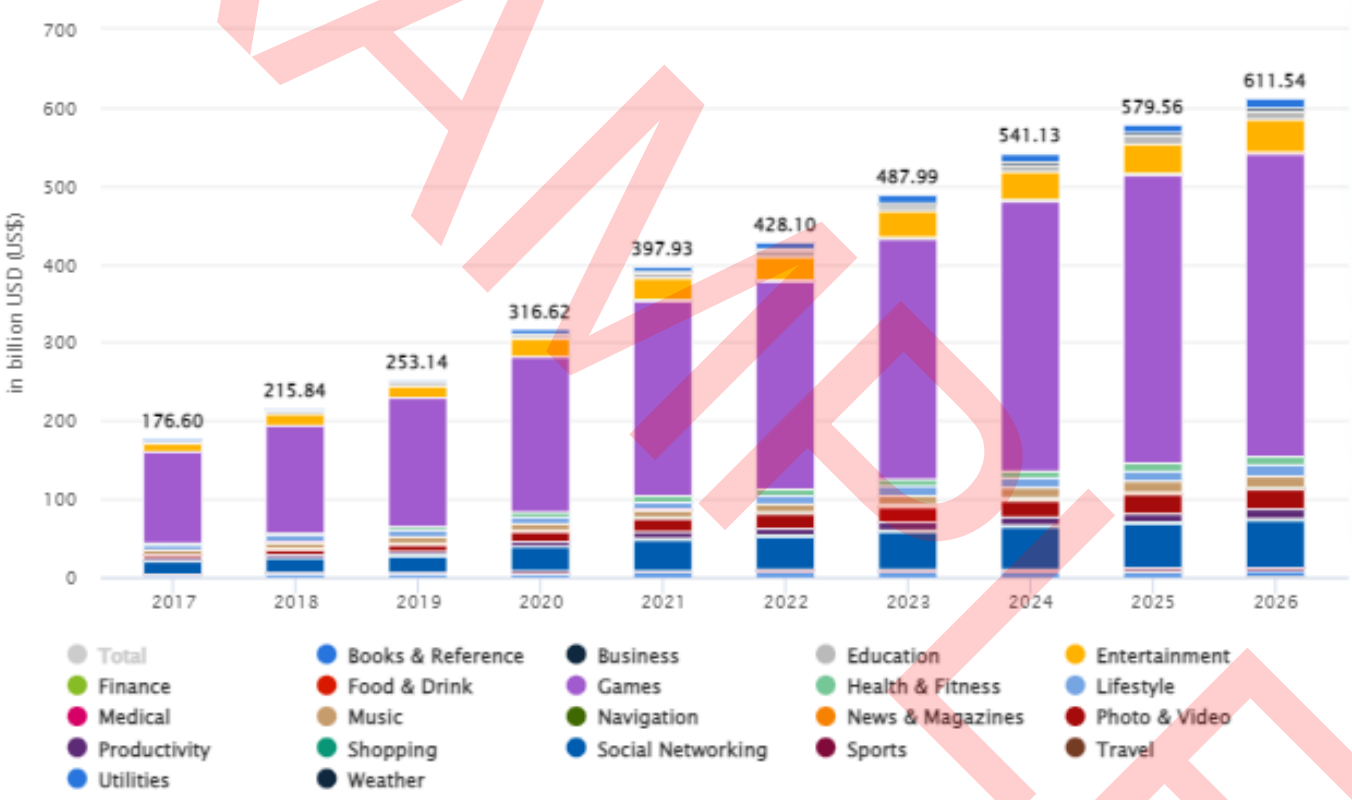


Figure 1: Apps revenue generation by segment

Source: Statista

The app market store occasionally uses certain algorithms to determine the most popular app categories, which reflect those where users are most engaging with the apps. As of the 3rd quarter of 2022 according to a statista report, some of the popular apps category occupying the foremost positions were the Gaming, Business, Education, Shopping, Health, and finance apps. In recent times,

finance apps category has been gaining more traction as their apps allow users to execute personalized financial operations like savings, investments, day-to-day buying, and spending-related activities from the comfort of their phones. Additionally, this is because start-ups have been significantly upending the financial industries by developing neo-banks and offering more accessible banking services to underserved communities. Recent report from the Thinkwithgoogle, also revealed that 73% of smartphone users use a finance app monthly to manage their finances. The Monzo app, a financial app, is one of the most widely used Neo Bank apps in the UK, with over 6 million users, according to a recently published Statista report.

Although great progress is being made in the mobile app development sector, not all developed apps have the same rate of adoption and continual usage over time. Previous research has been done to identify potential factors that can influence users' continuous use of mobile apps. Some of these are the works of Seok (2014), who used the motivational theory and the technology acceptance model to uncover characteristics that can influence users' persistent intention to use mobile apps and highlighted the ease of use, human connection, and social usefulness as their essential components. These determinant elements were also identified by Tam et al. (2020) using the theory of acceptance, use of technology (UTAUT2) model and the expectation confirmation model (ECM). As the primary predictors of app usage, they emphasized the role of habits, satisfaction, performance expectations, and effort expectations. Hence, identifying users' satisfaction and required modifications on apps' functionality are then core areas of focus for app developers to glean insights from app users' feedback.

This feedback is frequently dispersed over microblogging sites, social media platforms, and app stores, with the iOS and Google Play stores serving as the most popular locations for upload, distribution, and review gathering. These reviews, which are usually unstructured and number in the hundreds to thousands, often provide developers with the challenge of manually sorting through them to determine which problems are been raised and which should be prioritized for fixing.

In the past, app review analysis was conducted through manual processes, which had limitations of being laborious, slow, and ineffective. However, with the advent of recent technologies, these analyses are now being conducted through automated methods using machine learning algorithms. Some of its applicability has been in the areas of separating genuine reviews from fake ones, Martens et al. (2019), apps feature extraction Haroon et al. (2018), apps requirement mining, JHA et al.(2019), apps release planning, Ciurumelea et al.(2017), sentiment analysis of apps review, Ranjan et al. (2020), to mention a few.

Despite these developments, there hasn't been much attention paid to the field of app bug feature identification or digging further into these reviews to find recurring patterns, trends, and outliers for better judgment. Therefore, the purpose of this study is to close that gap by looking at app reviews to find problematic features that users frequently complain about, figuring out the feelings underlying those complaints, and visualizing patterns of feature complaints. In this study's limited view examination, the Monzo app will be employed, though their predictive abilities can be applied to other apps as well.

1.1 Problem Statement

App review analysis has become a growing area of focus in the app development ecosystem for its many beneficial reasons. A recent publication by statisa shows that as of February 2022 there was an average of 267 thousand reviews posted by global app users on the highest-ranking apps in the google play store, while there were 31 thousand reviews posted on the google play store across all app categories. As insightful as these may seem, these reviews are usually embedded with a mixture of real and spam reviews containing user experiences, bug reports, feature requests, and text ratings, Maleej et al. (2016). This hence creates a problem leaving app developers to filter through these reviews to get useful insights in the timeliest manner. Furthermore, because of the volume of reviews, there is a risk that certain feature requests will be overshadowed by other requests or irrelevant reviews. As a result, app users may waste time going through lengthy reviews to find out whether a problem has been fixed, while app developers may miss out on urgent requests for fixes.

1.2 Research Question

Building from existing work that has been done in app reviews analysis, this research work aims to add to the existing body of knowledge by seeking to answer the following research questions:

1. Which is the most suitable technique that can be used to analyse sentiment analysis from apps review?
2. Which insights can be gotten from topic modelling on apps review using deep learning models?

1.3 Aim

The aim of this project is to develop an automated apps review analysis system using a combination of deep neural networks and NLP technologies. Each review will be subject to sentiment analysis by the algorithm to categorize it as either positive or negative. The system also aims to extract topics from the reviews and their trend analysis on an interactive dashboard.

1.4 Objectives

The objectives for achieving the above-stated aim are outlined below:

1. Web scraping google play store and app websites to get user reviews on apps.
2. Using machine learning techniques to perform sentiment analysis on each review to classify them into positive and negative quadrants using predefined determinants.
3. Performing topic modelling using machine learning techniques and identifying app feature trends within given time interval.
4. Developing an interactive GUI interface that displays app features trend analysis contained in reviews within the given period.

The outline of this report is as follows: Chapter 1introduces the research problem; Chapter 2 provides a summary of the literature that has been written about the identified research problem; Chapter 3 defines the implemented model selection and metrics for their evaluation; Chapter 4 describes the employed implementation technique; Chapter 5 discusses the findings from the employed

implementation process; Chapter 6 provides a summary of the entire project work, outlining the limitations and recommendations for future works. Chapter 7 of the project provides the resources reviewed, while the project's appendix section contains supplementary materials used in the project.

2.0 Literature Review

Big data analytics has recently emerged as a key subject due to its benefits in improving products through analysis of consumer behaviour. Big data, which is routinely generated in large quantities, frequently requires mining to generate useful information. One of the 5 v's of big data, velocity, speaks to how quickly data is received, stored, and mined so that it is available when needed to make the best business decisions, resulting in economic success. Xu et al. (2015) expanded on this by conducting research and discovering that organisations who use a high level of traditional marketing analysis and big data analysis had the highest rate of new product success. Big data analysis for organisations frequently includes an examination of consumer feedback on the organization's products and services. With the advent of improved technology, more business now seek convenient and personalised ways through which customers engage with their services by developing mobile applications optimised for use on mobile devices.

2.1 Mobile Applications

These mobile apps hosted on apps marketplace store often provides its users with opportunities to post reviews on their experiences using the apps which serve as very rich source of information for app developers. These reviews come in high volumes daily, as previous research by Pagano et al. (2013) found that mobile apps received approximately 23 reviews per day and that popular apps, such as Facebook, received on average 4,275 reviews per day. Furthermore, a recent research publication by businessofapps(2020), a major media and information brand for the app market, 77% of people read at least one review before downloading a free app and 80% of people before buying a paid app. This makes apps review analysis a valuable tool or app developers to improve their users experience, customer satisfaction and enhance the overall performance of their app (Appentive 2021).

In recent time, mobile banking apps have transformed the financial business, owing to its ability to be used for carrying out financial and non-financial transactions using a mobile device. According to (Shaikh & Karjaluoto, 2015), mobile banking which refers to the execution of financial and non-financial transactions using mobile devices has received more greater attention from consumers lately as it offers high useability, usefulness and personalized banking experience. Furthermore, according to Federal Reserve research, "the use of mobile banking apps has expanded dramatically over the last decade, with about half of all smartphone users in the United States using a mobile banking app in 2019." (Federal Reserve, 2019). This trend is projected to continue, since mobile banking apps' convenience and accessibility make them an appealing option for consumers.

2.2 Traditional Techniques in App review Analysis

In the past, several techniques have been utilized to extract relevant information from apps reviews; the most popular of these is the use of traditional classification techniques. These techniques have been utilized in areas such as sentiment analysis, content analysis and network analysis for app reviews. For example, Liang et al. (2015), performed sentiment analysis to investigate the impact of app reviews on the success of mobile apps in the Android market. Another utilization of traditional approaches on app reviews is in content analysis, utilized by Nicholas, Jennifer et al. (2017) Shaw (2017) to identify common themes in apps review and patterns related to user retention. Also, Chen

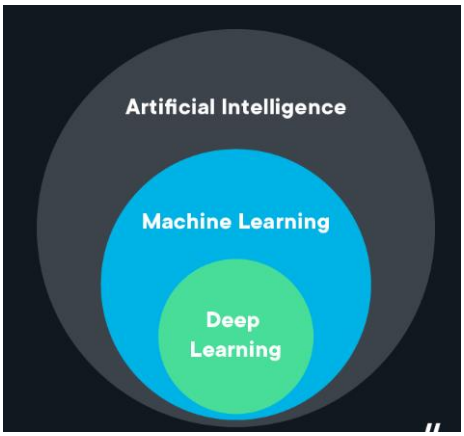
et al. (2018), employed network analysis to investigate the relationships between app producers, app users, and app reviewers in the Google Play Store using these traditional classification techniques.

Particularly in content analysis, Hassan et al. (2018) conducted an update-level analysis of app reviews using the traditional classification method. They determined that response time (100%), crashes (68%), network problems (60%), and functional issues (59%), had the most issues and the highest likelihood of occurring. Though their method could identify these issues, they mostly relied on manual labelling rendering it impractical for use over many apps' reviews.

In addition, Zhou et al. (2019) used Word2Vec and the Latent Dirichlet, both statistical modelling techniques, to identify user's needs from app reviews. Their methods classified the reviews into topics such as needs for functional enhancement, experience, and technology. Even though their method found several important areas for concern, it was not intuitive enough and required a lot of interpretation from subject matter specialist for understating. This drawback is highlighted further by Miroslav et al. (2022), who noted that using the traditional classification approach had several serious drawbacks, including the need for a labour-intensive manual process to label ground truth data (features), which was both time-consuming and inefficient. Considering improved methods to overcome these limitations, machine learning techniques are now being utilized in app analysis domain area due to their ability to provide a faster-automated process with a high level of prediction accuracy.

2.3 Deep Learning application in Apps Review Analysis

Deep learning, DL, a subset of machine learning trains a computer to perform human like tasks like speech recognition, virtual assistant services, self-driving cars, uses artificial neural networks with multiple layers of processing to recognise patterns and make predictions (Deng et al.(2014), Zhang et al. (2018), Kaluarachchi et al.(2021)). Due to its superior performance capabilities, its utilisation has recently exceeded that of machine learning techniques across a variety of domains such as cybersecurity, bioinformatics, robotics, natural language processing to mention a few (Alzubaidi et al(2021). Furthermore, as highlighted by Li Deng and Dong Yu (2014), certain significant areas that are currently benefiting from the recent research efforts of deep learning applications includes natural language and text processing, information retrieval, and multimodal information processing.



Source: Wikipedia

Figure 2: Branches of Artificial Intelligence

In apps review analysis for example, a branch of natural and text language processing, Lee et al. (2018).'s research highlighted the effectiveness of deep learning techniques for analysing app reviews using a neural network to predict accurately user's satisfaction. Furthermore, Suciati et al. (2020) compared the use of machine learning with deep learning algorithms for sentiment classification and rating prediction and discovered that deep learning algorithms performed better for both case scenarios.

There are however numerous deep learning techniques which has been used for analysing app reviews based on previous research, each with its unique capabilities and level of accuracy. From previous literature, some of these deep models includes: Convolutional Neural Networks(CNNs) commonly used for text classification tasks (Kim, Y. (2014), Li et al. (2017), Recurrent Neural Networks (RNNs) well-suited for modeling sequential data (Zhang, Z., & Zeng, D. (2017)),Transformer models, such as BERT and GPT (Kim, J., & Kim, S. (2019), He, Y., & Wang, S. (2020)), Hybrid models which combine multiple types of deep learning models together (Chen, D., & Liu, B. (2016, Li, H., Li, X., & Li, Y. (2018).

For the reason of their accuracy and effectiveness in app review analysis, deep learning techniques have hence shown promise in terms use, and they are likely to remain a popular option in the future despite some of the difficulties experienced when using them ((Zhang et al., 2018).

2.4 Sentiment Analysis in Text Classification

Sentiment analysis, a subset of Natural Language process refers to the process of recognising and categorising opinions from a piece of text and used to identify the writer's sentiments about a specific topic, product, or service (Liang et al(2015), Yadav et al.(2020)) . The aim of sentiment analysis is often to automatically extract and classify the emotional tone of a text, document, or speech, whether it be positive, negative, or neutral (Joshi et al.2016).

Sentiment analysis has been widely utilised by organisations to better understand their customers' perceptions of their services in a variety of domains to include customer services, market research, social media engagement, software development to mention a few. In general, there are three broad levels at which sentiment analysis can be classified: the document level, which analyses the sentiment of a document or a piece of text; the sentence level, which analyses the sentiment of individual sentences within a document; and the aspect level, which analyses the sentiment of features of a good or service. According to S. Mohammed et al. (2013), aspect-level sentiment is more fine-grained and useful in practise as it allows the extraction of more specific opinions and sentiments and provides detailed feedback on a particular product or service. Apps review analysis falls under the category of aspect-level analysis as it analyses the topic and aspects of an application.

There are however some limitations to performing sentiment analysis, one of key, which is the subjectivity of language, as the same words can be interpreted differently by different people as identified by Wang et al. (2012). In overcoming this challenge, many studies have been conducted to develop approaches to normalise and standardise the languages used in text data, including lemmatization, stemming, and the use of dictionaries to identify words and specific emotional connotations.

Some of the approaches that have been used to perform sentiment analysis includes lexical-based, rule-based, and machine learning-based methods. The lexical-based techniques use dictionaries and lexicons to detect words with positive or negative meanings and assign each word a matching sentiment score, while the rule-based techniques categorize text as positive, negative, or neutral using a set of pre-established rules. However, as identified by Krishnakumari et al. (2020), the lexicon-based and rule-based approaches still have the drawback that sentiments derived in one domain are likely to be interpreted differently in another domain. As a result, these authors proposed using machine learning/deep learning classifiers for performing sentiment analysis because these approaches can be adapted to any domain.

2.4.1 Deep Learning Models for Sentiment Analysis in Apps review analysis

Deep learning models have become very popular for use in apps review analysis in recent time. This usefulness is emphasized by Naseem et al. (2021), who identified them to be the most popular and effective approaches for review apps analysis due to their ease of use and high degree of accuracy in identifying patterns in the data and predicting the emotion of new text.

In identifying and classifying emotions contained in app reviews using deep learning, Al Wang et al. (2016) utilised and compared the performance of RNN architecture and LSTM model to classify sentiments of apps reviews as Positive, negative, and neutral. They found out that the LSTM model performed better with an accuracy of 87.8%. This claim is supported further by Zhang et al. (2019), who used the LSTM model and added extra features like the length of reviews and specific keywords to train the model to perform sentiment analysis on an apple play store data set. They found that the LSTM model outperformed conventional machine learning models.

Similar to this, a variety of deep learning models have been combined together to perform sentiment analysis on apps review analysis, and their performances have been compared. For example, Panichella et al. (2015) utilized machine learning methods combining Natural Language Processing, Text Analysis, and Sentiment Analysis to classify app reviews into the proposed categories and to identify sentiments contained in each review. Their results showed that a combination of the three techniques yielded a better result with (precision of 70% and a recall of 67%) as opposed to when each technique was used individually. Likewise, Khan et al. (2019) utilized a hybrid deep learning model, combining both CNNs and LSTM networks for sentiment of apps review and found that the hybrid model outperformed the individual CNN and LSTM models. Oladapo et al. (2020) used the Stochastic gradient descent, support vector machine, Random Forest, Logistic regression, and Multinomial naive Bayes method to analyse 88125 reviews of 104 Mental health apps from the Google Play Store and Apple Store to determine the sentiment present in them. With an F1 score of 89.42%, stochastic gradient descent was the model that performed the best. In their work, the polarity of the reviews' sentiments was divided into 21 negative themes and 29 positive themes.

Similarly, to how apps reviews are analysed for sentiment, they have also been utilised to determine the accuracy of star rating associated with each review. Luiz et al.(2018) buttressed this by utilizing Sentiment Lexicon and the Sentiment Identification (SACI) technique to analyse content of apps reviews. By dissecting, isolating, and evaluating individual contents the review, they identified inconsistencies of an overall star rating for an app, as the ratings often did not match the analysed emotions contained in the reviews, and hence they concluded that sentiment analysis is a better

measure for evaluation as it not only displays the reviews but also captures the emotions behind the comment given by the user. Additionally, Dhar et al. (2022) utilized a three-dimensional analysis to perform sentiment analysis on 146,914 reviews of two well-known payment apps; Pay and Phonepe that are used in India and found that sentiment emotions extracted from the reviews provided a more accurate star rating than emotions embedded in the text and reactions in the reviews' emojis.

Although the LSTM model has been used extensively in apps review analysis, it does have some limitations in that it is computationally intensive and may take a lot of time and resources to train, particularly for large datasets, as highlighted by Tai et al. (2015). Nevertheless, based on their use in prior studies, it has been discovered that LSTM models are more efficient than other deep learning models in extracting and classifying sentiments contained in apps reviews Zhang et al. (2019).

2.5 Topic Modelling in Text Classification

Topic modelling is a branch of text analysis that uses statistical techniques to automatically group comparable terms into related categories after scanning documents for potential words and phrases. Its usefulness is also in identifying hidden topics and themes based on nonrelated words in a document. The application of topic modelling has been used in information retrieval, document classification, and text summarization (Cao et al.(2015), Liu et al.(2016), Asmussen et al.(2019)).

Several techniques have been employed for topic modelling to include: Latent Dirichlet Allocation (LDA), Non-Negative Matrix Factorization (NMF), and Top2Vec, machine learning techniques to mention a few with each having advantage over the other. Due to its ease of use and capacity for handling huge datasets, the LDA has been previously identified as the preferred topic modelling algorithm although it has the drawback of assuming that every word in a document is equally significant, which isn't necessarily the case as identified by Yang et al. (2015). The NMF, though more flexible than the LDA as it allows for the incorporation of word frequencies and document metadata in topic modelling, has the disadvantage of being more computationally expensive and hence not suitable for use on large datasets. The Top2Vec, a variant of the word2Vec algorithm uses a neural network to learn the relationships between words but focuses on modelling the relationships between words. It however does not consider the broader context of the document thereby making it difficult to identify the main topics in a collection of documents when it is used alone (Gan et al. (2016). This shortcoming hence makes it unsuitable for topic modelling, as it requires the use of better techniques.

In recent times, machine learning techniques have however proved a better approach for topic modelling as identified by Bai et at.(2018), as they automatically uncover the underlying themes in a collection of texts through the use of word term frequency and metadata from a large collection of document.

2.5.1 Deep Learning Models for Topic modelling in Apps review analysis

The application of topic modelling in apps review analysis has been very beneficial to both app developer and users. For users, it provides a simplified way by which they can get a fine-grained understanding of specific working functionalities of apps by reading reviews posted by other users. In the same vein, it enables developers know specific aspects of their apps that are dysfunctional that users are most concerned about for maintenance and marketing optimization purpose.

Earlier works done on topic modelling from apps review was conducted by Guzman and Maalej (2014) by employing the Latent Dirichlet Allocation (LDA) techniques with the weighted-average techniques. With their approach, they were able to extract and aggregate fine-grained explicit topics contained in the reviews into high-level features. By analysing the extracted topics with relevance to the app, they found that the top 10 extracted topics typically described real app features mostly engaged with by users.

In identifying trends of hot issues encountered by users of mobile apps, Cuiyun et al. (2015) created and used an AR Tracker for topic identification and review-ranking in prioritizing user reviews. Their method employed and contrasted between Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA), Random Projection (RP), Non-Negative Matrix Factorization (NMF), and LDA Gibbs Sampling techniques. After analysis, it was discovered that the AR Tracker, when used with either of the topic identification techniques, had a higher level of success extracting these topics. Its method was also advantageous because it didn't require manual labelling of the topics before the filtering process, which made it better suited for modelling on large set of reviews.

Ciurumelea et al. (2017) analysed app reviews for code mapping using User Request Reference system which leverages on a combination of decision tree sequences and Gradient Boosted Regression Trees (GBRT) to create predictions by integrating the results of several decision trees. The system classified the analysed 7754 reviews into a list (17 + 1 for Complaint) high- and low-level categories and proved to reduce the reduce the time needed to manually analyse reviews by up to 75%.

Palani et al. (2021) extracted topics from short text sentences by analysing 40,000 tweets from Twitter without using search terms or specific topics of interest. They used a T-Bert model, which combines the LDA model and BERT model, and they compared its performance to using only the LDA model. They found that the T-BERT model performed better than the LDA model, which had an accuracy level of 90.81%, but they also noted that the model had some limitations, including an inability to handle complex data aspects, linguistic expertise, and semantic search as found in tweets.

Al Moubayed et al. (2020)'s innovative framework for text classification used topic modelling by combining Bag of Words, SVM, Lexical, Glove, LDA, and LSTM for both feature extraction and topic classification. These models were applied to 10 benchmark datasets for spam filtering, topic modelling, and sentiment analysis of movie reviews. Their method was deemed effective since it could uncover structural relationships across the topics while also incorporating their contextual relationship.

Based on these reviewed research, deep learning techniques have hence proved to be better suited for topic modelling especially on large dataset, with high level of accuracy. However, as identified by Gülle et al. (2020) , their applicability could be computationally expensive to use, often requiring manual labelling of the extracted topics is still necessary to fully understand the discourse contained in each topic.

In concluding this chapter, it began by highlighting from previous research various techniques and emerging trends that are being utilized in sentiment analysis and topic modelling of apps review whilst examining the strength and weakness of each technique. This project thus contributes to the

body of knowledge by applying and analysing some deep learning models with strong focus on LSTM and BERT model for apps analysis, furthermore, highlighting insights that can be gotten using these models.

3.0 Model Selection and Evaluation

This project aims at using deep learning techniques to categorising sentiments contained in the reviews as well as clustering and extracting topics contained in them. The selected models to achieving these tasks are the Latent Dirichlet Allocation (LDA) model, Long Short-Term Memory (LSTM) Model, the Bidirectional Encoder Representation from Transformers (BERT) model, Support Vector machine (SVM), Logistic regression (LR) and Multinomial Naïve Bayes Algorithm (MNB). These models were selected based on results from previous research and their performance for achieving the project's objective swill be evaluated against each using certain evaluation parameters.

3.1 Long Short-Term Memory Model (LSTM)

The LSTM model is a deep learning architecture built on an artificial recurrent neural network (RNN) commonly used in natural Language Processing tasks(NLP) that handles long-chained task sequences and recalls crucial information as it does so. Discovered by Hochreiter and Schmidhuber in 1997, LSTM address the issue of vanishing gradient problems that was encountered in training traditional Recurrent Neural Network. It employs a gating mechanism using its three gates, input, output, and forget gate, to control the flow of information through its network. Additionally, it has two states:, the Cell state for remembering long term memory information and Hidden state for the short term memory information.

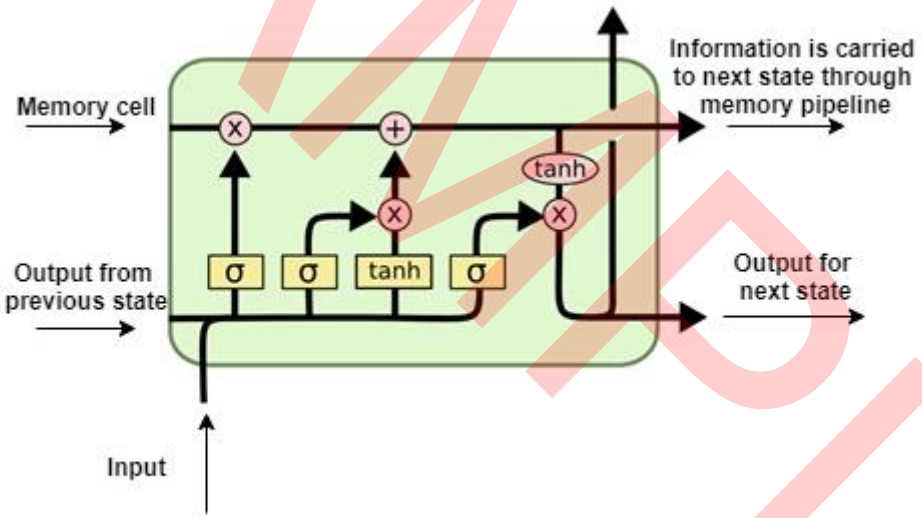


Figure 3:LSTM Network

Source: COLAH

Within the forget gate, information from the current input X(t) and hidden state h(t-1) are passed through the sigmoid function which generates values between 0 and 1.

Furthermore, the internal working of LSTM model uses the tanh function to regulate the input values between -1 and 1 and produces an output vector (C~(t)). The cell state is updated by multiplying the previous state value by the resultant output vector. If the outcome is 0, then values will get dropped in the cell state. If not, the network performs a point-by-point addition to the previous cell state to produce a new cell state. As the model runs, values which are 0 are dropped, and values closer to 1

are passed to the next input gate. For each sequential step, the input state updates the cell state with new input values in addition to values from the forget gate.

Due to its backpropagation abilities, the LSTM model is extremely efficient in apps review analysis, as identified by Zhang et al. (2017) as it recalls and maintains the presence of certain words (features) in the input stream as the model goes through a series of steps.

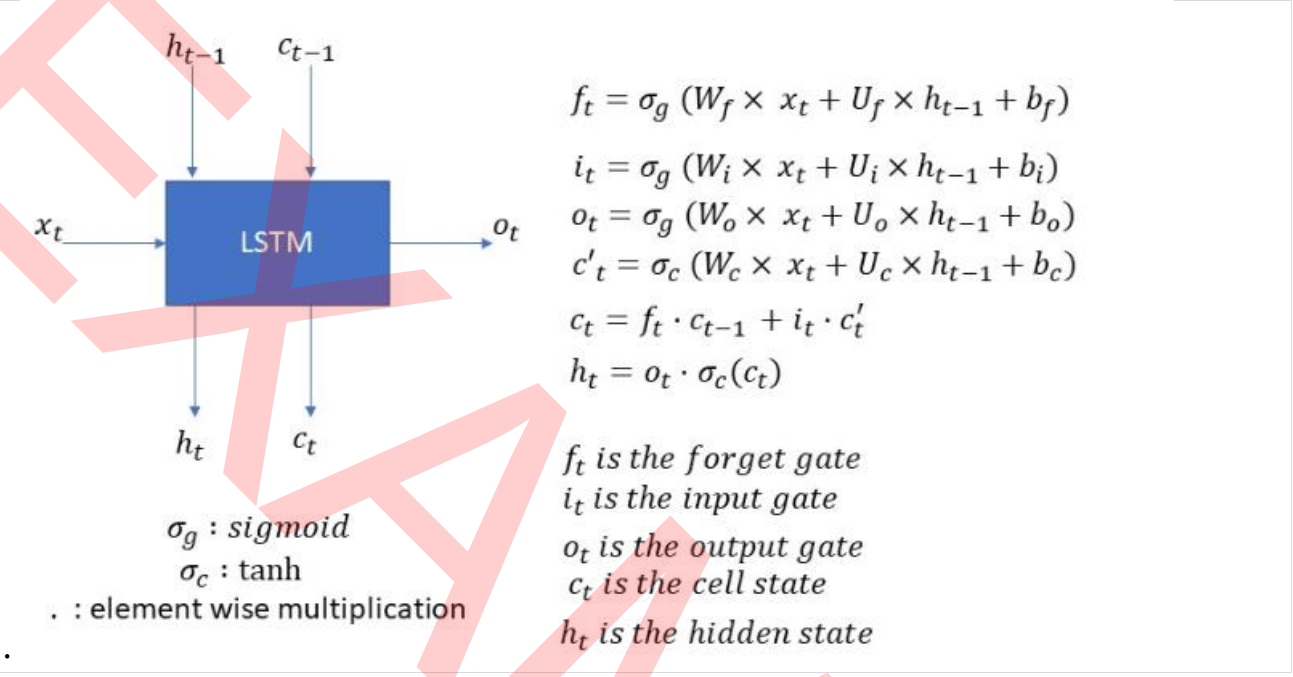


Figure 4:LSTM Model Architecture

Source : TowardsAI

3.2 The Bidirectional Encoder Representations from Transformers Model (BERT)

The BERT model is a pre-trained model created in 2018 by Google researchers widely used for a variety of natural language processing tasks and language translation. Its application in apps review analysis has been used by previous research such as Chung et al., (2020) who used it to classify apps reviews as positive and negative sentiments and identified it to outperform other machine learning models with an accuracy of 89.5%. The BERT model has large number of encoder layers (Transformer Blocks) and feedforward networks. It takes in sequence of words as input moving up the stacks and uses self-attention at each layer and feeds forward the results to the next encoder through the network to the next encoder.

The BERT's model has advantage over other models as it learns contextual associations between words in a text, by employing Transformer, an attention technique which runs in parallel during implementation, making it faster than the LSTM which learns sequentially as identified by Chen et al., (2019). Furthermore, it has the advantage of evaluating several properties of words, such as semantics, syntax, and vocabulary, as well as their position in a sentence using bidirectional architecture. The BERT model is primarily helpful for tasks involving natural language processing, like apps review analysis, as identified by Devlin et., (2019) because it is computationally resourceful paying attention

and only remembers words in a sentence that are important for subsequent processing by establishing differential weights for words in sentences.

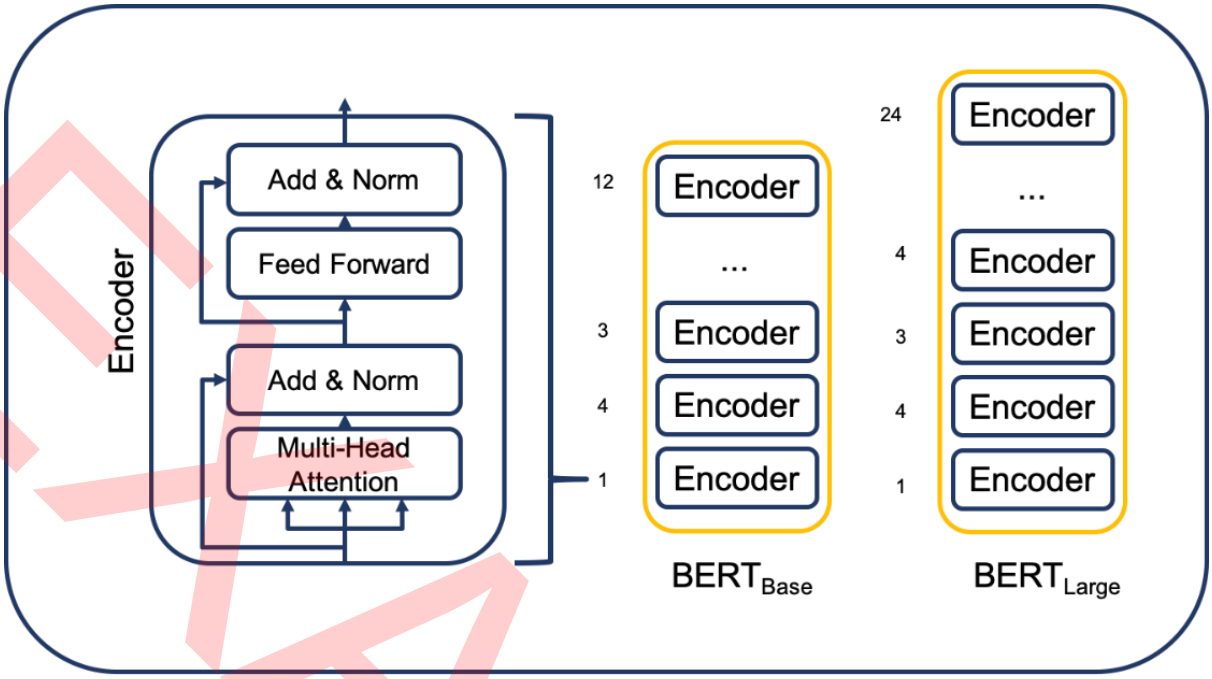


Figure 5: Architecture of BERT Model

Source: Analytic Vidhya

3.3 The Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation is a technique used for topics topic modelling in text mining and natural language processing tasks. Introduced first by David Blei(2003), it uses the Bayesian approach to estimate the probability of topics and words contained in a document. With advantage of being able to handle large and sparse data, the LDA model has been used severally in the text mining domain such as information retrieval (Deveaud, et al.2014), text summarization(Allahyari et al., 2017), sentiment analysis (Wang et al.2011). In apps review analysis, the LDA model, it has been used severally by researchers to extract insights about what apps users are saying as contained in the extracted topics. It has major advantage of been able to automatically identify the topics contained in reviews without manual annotation of the data, and hence its suitability for information retrieval from reviews.

3.4 Supervised Learning Models

Supervised learning models such as Support vector Machine (SVM), Linear Regression (LR), Multinomial Naive Bayes Algorithm(MNB) are widely used for regression and classification problems. These models utilize labelled data to be to be used to make predictions. They have been widely used in app review analysis for making predictions, especially in sentiment analysis, to identify emotions contained in the reviews. Over the years, and as earliest utilised for apps review analysis, these modes have achieved significant performances when utilized for classification tasks.

3.5 Model Evaluation

The evaluation of a machine learning model is a crucial stage in any machine learning assignment since it enables measurement of the model's efficiency in producing precise predictions or classifications on new data. To this end, specific metrices will be applied to evaluate the effectiveness of the utilized model that this project implementation. These metrics were selected based on research and industry-acceptable standard.

3.6 Evaluation Metrices for the Models

The metrics used in evaluating the models are outlined below:

1. **Accuracy**: The accuracy score is a commonly used evaluation metric for machine learning models, as it is very straightforward for use. For use in this thesis, the accuracy score will be the measure of the percentage of app reviews that the models correctly classifies into the positive and negative sentiment quadrants. This is done by comparing the predicted labels produced by the models' model with the ground truth labels (i.e., the true labels of the app reviews). The number of correctly classified app reviews is then divided by the total number of app reviews, and the resulting value is multiplied by 100 to get the accuracy score as a percentage. It is a useful evaluation metric because it is simple and easy to understand and gives a clear picture of the overall performance of a model. However, as pointed out by Alessia, et al., (2015), it has a limitation of not considering relative importance of correctly classifying different sentiments of reviews as it may be biased to a particular sentiment.

2. **F1 Score**: The F1 score is the harmonic mean of precision and recall and is often used as a single metric to represent the overall performance of a model. Precision measures the proportion of predicted positive instances that are positive, while recall measures the proportion of actual positive instances that are predicted as positive. A high F1 score indicates that the model performs well at both identifying positive app reviews and correctly classifying them. The F1 score can be calculated using the formula: F1 = 2 * (precision * recall) / (precision + recall). The F1 score although very useful for text classification task, however, has a limitation of being sensitive to imbalanced class distribution as it may be biased toward the prevalent class in the dataset.

3. **Coherence Score (CV):** The coherence score is an evaluation metrics that measure how well topics fit together semantically. This also measures how interpretable and consistent topics generated by the machine learning model is. A high value coherence score shows that the words contained in a topic generated by a model is more semantically related and is often used to evaluate the quality of topics generated by models when comparing them with each other.

4.0 Implementation Technique

This chapter outlines the steps resources and technique employed for implementing this project. It includes the research steps, data resources, software tools, development environment, and timeline required for the project implementation. The process flow for the implementation is shown in Figure 6 .

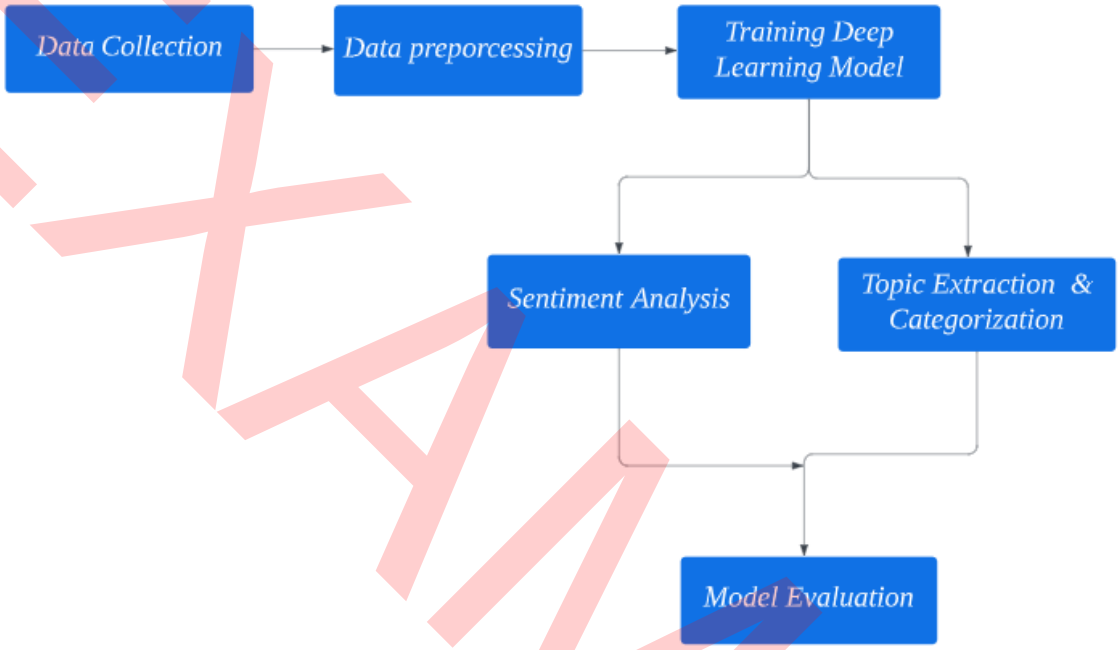


Figure 6:Project Implementation Process

4.1 Project Management Plan

The project implementation process began with the creation of a project management plan outlining the activities and duration for the full project implementation process spanning over a three-month period. This is shown in Figure 29 at the Appendix section.

4.2 Project Resources

Pycharm and Google collab were used as the integrated development environments for data preprocessing, visualization, model training and evaluation using the imported modules from Python as outlined in Table 1. Pycharm was used for sentiment analysis, while google collab was used for the topic modelling as it was more suitable for installing the deep learning models.

Tools		Use functionality
Python 3.8+		Software development
Google Collab		Software development
Github		Version control
Python Libraries	Google Play Scraper	Web scraping apps review from Google play store
	Pandas and NumPy	Data preprocessing and wrangling
	NLTK	Text dataset processing
	Matplotlib	Data visualization
	TensorFlow	For processing and loading data
	BERTtopic	For implementing the model
	Keras	For implementing the Neural Network
	Scikit-learn	Model training and implementation
	Transformer	Model training and implementation
	Streamlit	Building the Dashboard

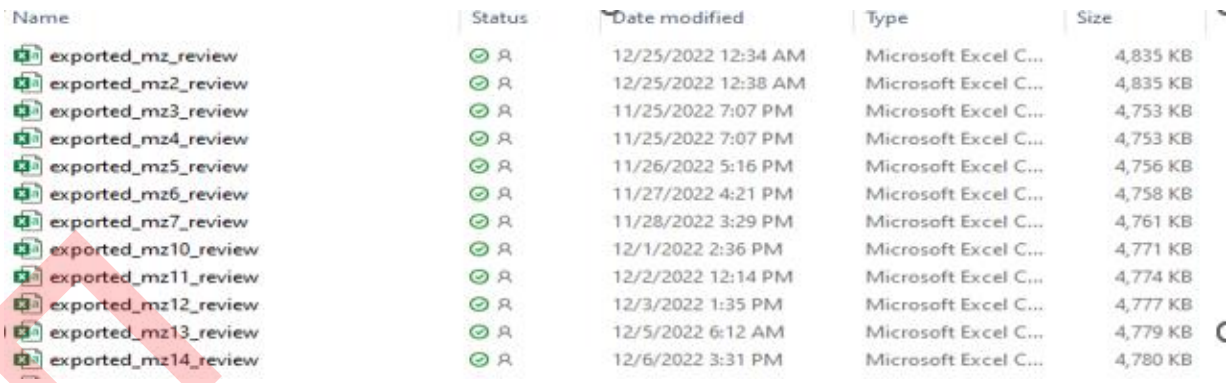
Table 1: Software tools used for Project implementation

4.3 Software Implementation Process

This process involved all the activities that was undertaken to pre-process the data, train the machine learning model for performing sentiment analysis, topic modelling, and evaluating the model's performances.

4.3.1 Data Collection

A secondary data collection method was employed for collecting historical apps review of the Monzo app from google play store through the web scraping technique. The web crawler scraped the google play store for the period of 22 days, spanning a period from 2nd November to 24th December 2022. The data was collected daily and stored in individual csv files as shown in Figure 7.



Name	Status	Date modified	Type	Size
exported_mz_review		12/25/2022 12:34 AM	Microsoft Excel C...	4,835 KB
exported_mz2_review		12/25/2022 12:38 AM	Microsoft Excel C...	4,835 KB
exported_mz3_review		11/25/2022 7:07 PM	Microsoft Excel C...	4,753 KB
exported_mz4_review		11/25/2022 7:07 PM	Microsoft Excel C...	4,753 KB
exported_mz5_review		11/26/2022 5:16 PM	Microsoft Excel C...	4,756 KB
exported_mz6_review		11/27/2022 4:21 PM	Microsoft Excel C...	4,758 KB
exported_mz7_review		11/28/2022 3:29 PM	Microsoft Excel C...	4,761 KB
exported_mz10_review		12/1/2022 2:36 PM	Microsoft Excel C...	4,771 KB
exported_mz11_review		12/2/2022 12:14 PM	Microsoft Excel C...	4,774 KB
exported_mz12_review		12/3/2022 1:35 PM	Microsoft Excel C...	4,777 KB
exported_mz13_review		12/5/2022 6:12 AM	Microsoft Excel C...	4,779 KB
exported_mz14_review		12/6/2022 3:31 PM	Microsoft Excel C...	4,780 KB

Figure 7: Scrapped Data store in individual CSV Files

The data was loaded into Pycharm using pandas library where the data pre-processing was conducted done.

```
# Load the dataset of app reviews
df = pd.read_csv('Monzo.csv')
df.head(5)
```

	reviewId	userName	userImage	content
0	129510a0-b042-45f4-9f0a-3c497014652d	Jose Teles	https://play-lh.googleusercontent.com/a...	The only problem is when you ca
1	1616dc35-a879-43c6-bc7b-183388e4cde2	Manchu_uk	https://play-lh.googleusercontent.com/a...	It takes age's to receive card'
2	d5f8ad28-78ae-446f-85d1-f6c1a2d1f0ef	Patient Onyealusi	https://play-lh.googleusercontent.com/a...	I've not gotten my Acc
3	e219a36b-cad0-4eed-bce3-6687587d0efc	mutungi marvin	https://play-lh.googleusercontent.com/a...	I was denied an account with no
4	ad2dd2fd-67fc-4423-bf94-15a2f98cff86	M Absar Asif	https://play-lh.googleusercontent.com/a...	Not a good experience with the

Figure 8:Loading the Dataset on PyCharm

The features of the dataset is listed in described in Table 1.

Feature	**Description**	**Data type**
Review ID	Unique Identifier for each review	Object
User name	Name of the reviewer	Object
User image	Image of reviewer	Object
Content	The apps review	Object
Score	Numerical rating of the review	Integer
Thumbs up count	Count of thumbs up given for each review	Integer
Review created version	The version of app being reviewed	Object
At	When review was posted	Object
Reply content	Reply to the review	Object
Replied At	When reply was posted	Object

Table 2: Features of Monzo Apps review Dataset

4.3.2 Data Cleaning

The Data pre-processing required first required getting unique reviews from the dataset as the daily collected reviews had duplicate values from previous day collection. To achieve this, two options were explored. The first was done to join all the csv files into a single csv file using python script, before further pre-processing. The other method involved taking reviews for the last day collection date and prepressing it using python scripts. The later method was eventually used for the project implementation as it proved technically better.

In exploring the consolidated data set, it contained 15,031 rows and 10 columns (features), of which the two columns(features) needed for training the model are the content and score features. The dataset also contained several duplicates and 1 null value in the content feature, 5 null values in the score feature. These identified values were dropped as they would not have significant impact in training the model but rather serve as noise. This hence reduced the dataset to 12,863 rows and 2 features.

```
topic_df.shape
```

```
(12864, 2)
```

```
#Checking for null values from the new dataset
check_nan = topic_df[topic_df.isna().any(axis=1)]
check_nan
```

	content	score
4662	NaN	5

1 rows × 2 columns Open in new tab

```
#Dropping null values from the new dataset
topic_df = topic_df.dropna()
```

```
#Checking the shape of the dataset after dropping the null values
topic_df.shape
```

```
(12863, 2)
```

Figure 9:Cleaning the Data

4.4 Implementing Models for Sentiment Analysis

Sentiment Analysis was conducted on the data set using the LSTM and BERT model which included the following pre-processing steps:

- **Creating a new column**: A new column was created called sentiment which identified the various sentiment polarity, 0 as negative, and 1 as positive using the score rating. This was then used for the model implementation to identify and extract the actual sentiments contained in the reviews.

```
#Printing from the new dataframe
topic_df.head(5)
```

	content	score	sentiment
0	The only problem is when you cash out ,...	3	1
1	It takes age's to receive card's, i hav...	1	0
2	I've not gotten my Account Number	3	1
3	I was denied an account with no explana...	1	0
4	Not a good experience with the app. And...	1	0

Figure 10: Creating new Sentiment Column

- **Tokenizing the content**: This was done to split the words contained in the content feature into tokens to helps the model interpret the meaning of the texts by analysing the sequence of words.
- **Removing the stop words and punctuations**: Words such as **!, ‘’ ’’, .,** which do not carry any useful information was removed from the text for enable the model focus on only relevant words during execution.
- **Lemmatizing the words**: The NLTK package was used to achieve this to derive the root words in the sentences before training the model.

```
#Tokenizing, removing stop words, lemmatizing
def preprocess_df(df):
    # Create a new column to store the processed text
    df['processed_text'] = ''

    # Create a tokenizer and stemmer
    tokenizer = PunktSentenceTokenizer()
    stemmer = PorterStemmer()
    lemmatizer = nltk.WordNetLemmatizer()

    # Get a list of English stop words
    stop_words = set(stopwords.words('english'))

    # Iterate over the rows of the DataFrame
    for index, row in df.iterrows():
        # Tokenize the text
        tokens = tokenizer.tokenize(row['content'])

        # Lowercase the text
        tokens = [token.lower() for token in tokens]

        # Remove punctuation
```

Figure 11:Data Pre-processing

4.4.1 Training and Evaluation of LSTM Model

Training of the LSTM model included setting up and tuning the hypermeters to obtain the best values for accuracy. The values utilised in this project were obtained by fine-tuning the hyperparameters to get the combination that produced for the model's best accuracy value. A batch size 64 was used to determine the amount of training data that must be processed before the model's internal parameters are changed.

- Sigmond activation function was used to facilitate the learning of non-linear relationships by the neural network.
- The word length for the review was padded to a maximum length of 150 to ensure the input texts ad the same length.
- The train size of 80% was used, and the text size of 20% of the data was used.
- A drop out layer of 20% was used, with an Adam optimizer, with accuracy metrics used to calculate the loss function.

```
# Build the model
model = Sequential()
model.add(Embedding(max_words, 50, input_length=max_length))
model.add(LSTM(64, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 12: Setting Hyperparameter for LSTM Model

```
# Train the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=5, batch_size=32)

Epoch 1/5
322/322 [==============================] - 109s 309ms/step - loss: 0.3431 - accuracy: 0.8474 - val_loss: 0.2294 - val_accuracy: 0.9098
Epoch 2/5
322/322 [==============================] - 138s 429ms/step - loss: 0.1773 - accuracy: 0.9330 - val_loss: 0.2373 - val_accuracy: 0.9075
Epoch 3/5
322/322 [==============================] - 155s 482ms/step - loss: 0.1289 - accuracy: 0.9540 - val_loss: 0.2459 - val_accuracy: 0.9087
Epoch 4/5
322/322 [==============================] - 128s 398ms/step - loss: 0.1008 - accuracy: 0.9659 - val_loss: 0.2757 - val_accuracy: 0.9106
Epoch 5/5
322/322 [==============================] - 116s 359ms/step - loss: 0.0855 - accuracy: 0.9711 - val_loss: 0.3106 - val_accuracy: 0.9036
```

Figure 13: Training the LSTM Model

The model was successfully classified the revies into positive and negative and its evaluation on test data achieved 90.4% accuracy score, and 93.47% F1 score.

```
#Getting the Accuracy score
print('Accuracy: %.3f' % accuracy_score(y_test, predictions))
```

```
Accuracy: 0.904
```

```
#Getting the F1 score
print('F1 score: %f'  % f1_score(y_test,predictions))
```

```
F1 score: 0.934771
```

Figure 14:Evaluating LSTM Model's Performance for Sentiment Analysis

4.4.2 Training and Evaluation of Machine Learning models

The processed data was fed into the machine learning models namely Support vector Machine (SVM), Linear Regression (LR), Multionomial Naive Bayes Algorithm(MNB) for predicting the sentiment analysis on the dataset. By using a split training and test size of 80% and test size of 20% respectively, and setting the random state to 42, the model's accuracy and F1 score obtained are:

Model	Accuracy	F1
SVM	91%	82%
LR	91%	81%
MNB	88%	72%
LSTM	90.4%	93.47%

Table 3:Evaluating performance for Models for Sentiment Analysis

4.5 Implementing Models for Topic Modelling

The two models used for topic modelling are the LDA model. The BERT Model which is a pretrained used an unsupervised learning approach to identify topics contained in the on the plain text (corpus). Prior to model training, the data was cleaned by removing null and duplicate values as outlined in the data cleaning process above. Also, the following preprocessing steps were done on the data before loading it into the model to identify topics from the reviews:

- **Tokenizing the content**: The BERT converted the cleaned reviews into tokens that before loading it into the model
- **Fine tuning**: The model fine-tuned the review content by learning specific language patterns contained in the review using its transformer capability to understand context of word and ambiguity in language by referencing surrounding words to it.

Putting the content feature into a list

```
[19] docs = list(topic_df.loc[:, "content"].values)
```

Priniting a part of the content list

```
docs[3:]
```

```
 'Very nice app and easy to access',
 'excellent bank perfect',
 'I do not have experience',
 "Looking to download monzo? Download Revolut instead, it's genuinely significantly better!!",
 'Very very very very bad account I try 3 times but its not open my account',
 'Excellent service',
 'Absolute first class online banking application which I really love using. Monzo just make everything so easy, highly recommended. Easy 5 stars.',
 "it's a good",
 'I love it',
 'Love monzo so easy to use.',
 "I'm from Nigeria and I am in need of the monzo loan severely so could start up a good business but I have been finding it difficult to register,,,
it",
 'Please is this app granting loan? And also did I need to pay some money before I could get or withdraw the loan granted to me?',
 'Slow in responding to issue and mail',
 'So its day 5 and it still hasnt finished thinking about allowing me an account ..heard so much good stuff about monzo but thb im thinking its not
day and age !!!!',
```

Figure 15:Snippet of Apps Review content

4.5.1 Training and Evaluating the BERT Model

The model identified each topic by gathering a collection of words with the same semantic meaning into a cluster, using the KMeans clustering approach, and the reviews (document) were distributed across the extracted topics. Each topic (cluster) contains words which are weighted using the TF-IDF score to get the most frequently occurring words contained in the topic, and the model arranged and ranked position the size of each topic in descending order based on the word count contained in them. Although the majority of the 168 topics that the model BERT model automatically identified had similarly recurrent phrases in each of them, the topics were nonetheless reduced and clustered together using the model's hierarchical clustering function.

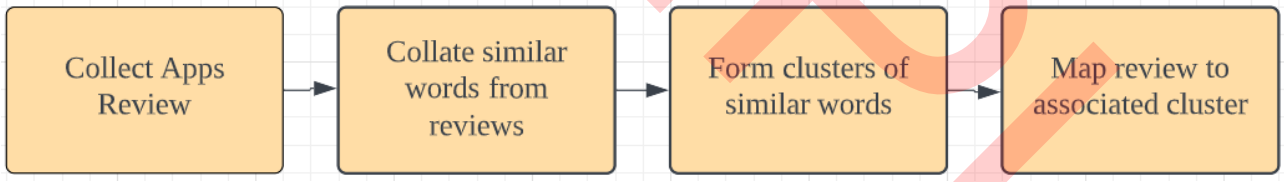


Figure 16:Process flow for Topic Modelling

```
[22] model = BERTopic(language="english")

topics, probs = model.fit_transform(docs)

Downloading: 100%  1.18k/1.18k [00:00<00:00, 52.5kB/s]
Downloading: 100%  190/190 [00:00<00:00, 6.91kB/s]
Downloading: 100%  10.6k/10.6k [00:00<00:00, 368kB/s]
```

Figure 17:Training the model for Topic Extraction

```
model.get_topic(2)

[('manage', 0.023673638058943264),
 ('track', 0.0229003581082511),
 ('money', 0.022129840421890517),
 ('way', 0.02186591490545888),
 ('finances', 0.020184804154379683),
 ('easy', 0.019076000857111326),
 ('spending', 0.017033333357676518),
 ('budget', 0.015465670672142719),
 ('helps', 0.01470926021835162),
 ('expenses', 0.013928871518093662)]
```

```
model.get_topic(9)

[('fingerprint', 0.039318125207087724),
 ('security', 0.03227626587113765),
 ('pin', 0.02202645271630736),
 ('password', 0.019098215335503772),
 ('finger', 0.018785865041315044),
 ('print', 0.015197588571055768),
 ('access', 0.012693828422805654),
 ('scanner', 0.012304883275724446),
 ('phone', 0.012071705063249567),
 ('biometrics', 0.011652896919305338)]
```

Figure 18:Extracted topics with common words

4.5.2 Visualizing the extracted Topics

Having identified the extracted topics, various visualization technique was employed to further identify meaningful insights from them which includes the word scores, word cloud similarity matrix and word cloud displayed in fig 22, fig 23 and fig 24.

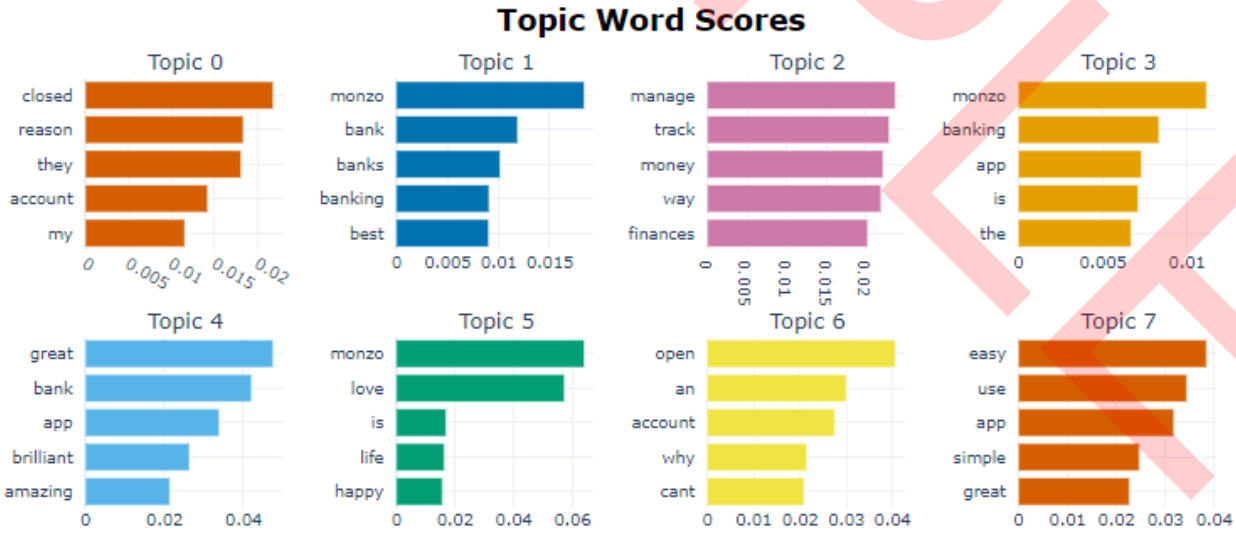


Figure 19:Word score showing word frequency per topic

Using the word cloud from each of the extracted topics, these were further manually labelled to give more context on each topic as displayed in *Table 4*.

Topic No	3 Most Frequent words	Manually Labelled Topic
Topic -1	Is, it, to	Not descriptive
Topic 1	Closed, Reason, App	Account closure related Issues
Topic 2	Monzo, Banking, app	Not descriptive
Topic 3	Monzo, love, happy	Not descriptive, but reveals positive apps remarks
Topic 4	Track, Budget, finance	Budgeting feature
Topic 5	Account, money, closed	User's closure related feature
Topic 6	Brilliant, Fantastic, excellent	Not descriptive, but reveals positive apps remarks
Topic 7	Easy, Simple, great	Apps ease of use
Topic 8	Security, Fingerprint, password	Apps' security
Topic 9	ffx, tx, este	Not descriptive
Topic 10	download, update, install, fax	Upgrade
Topic 11	Video, Camera, Photo	Camera, Audio & Video related issues

Table 4:Manually labelled extracted topics using BERT model

Furthermore, in identifying the similarity between the topics, the similarity Matrix, Hierarchical map and intertopic distance map were used. The similarity matrix shows the text similarity between two documents(reviews) contained in the dataset.

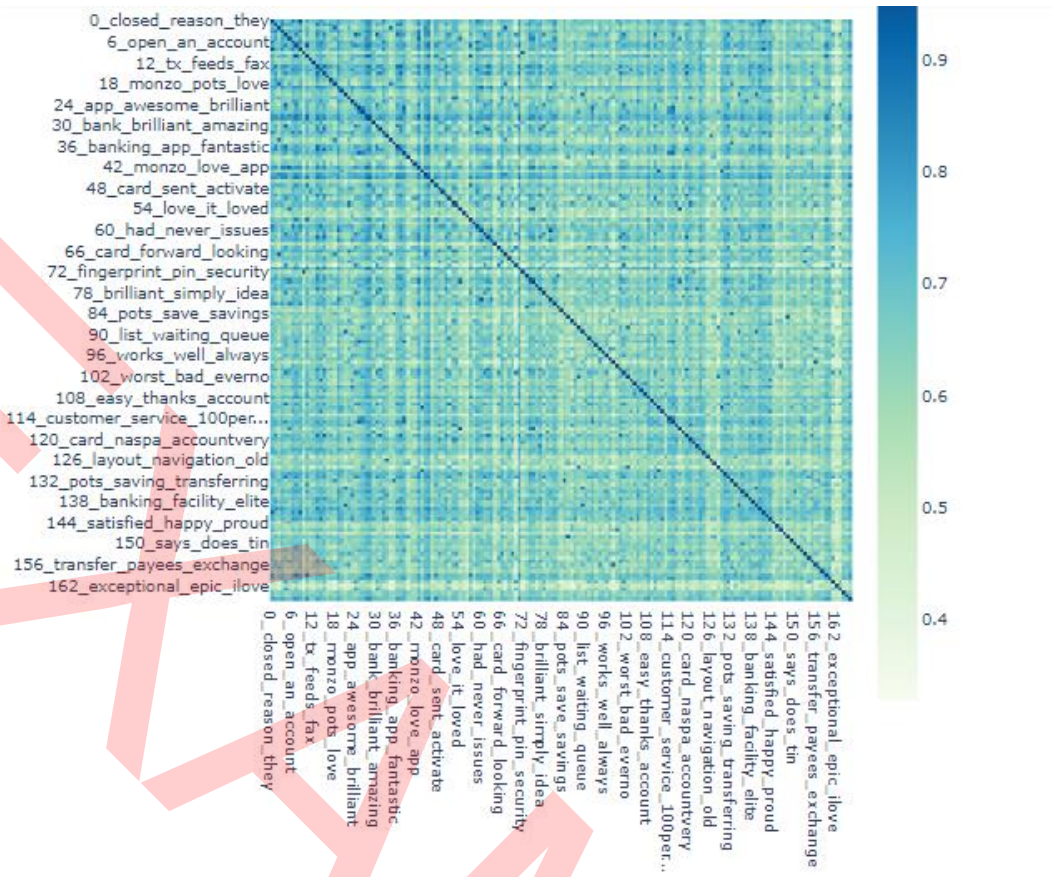


Figure 20:Similarity matrix between reviews

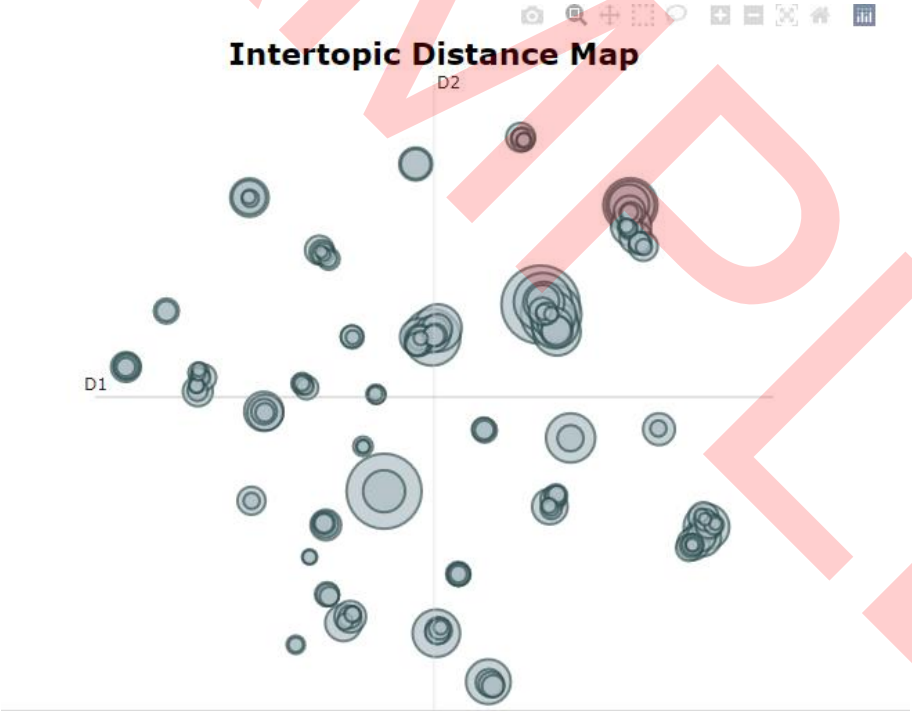


Figure 21:Intertopic Distance Map

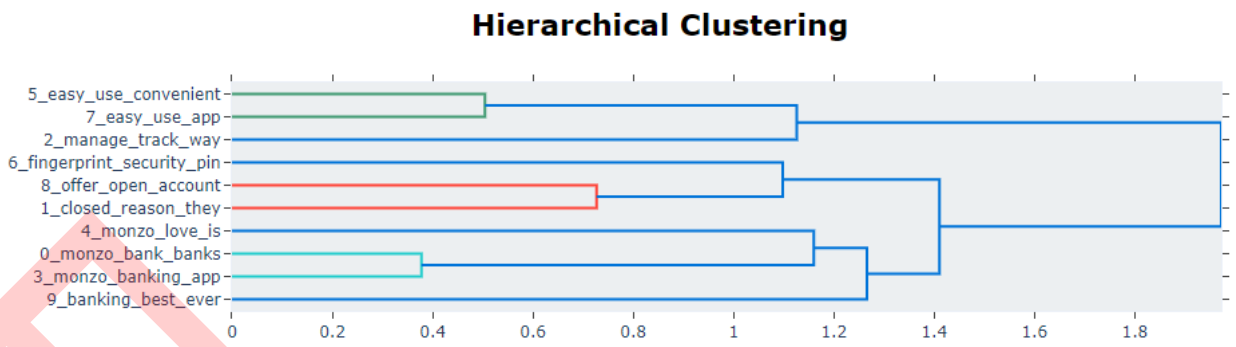


Figure 22:Hierarchical Clustering of top 10 Topics

For the top 10 topics, the hierarchical clustering of the topics shows that they contain similar words and hence can be grouped together. From the top 10 topics, topics 5 & 7, topics 7 & 2, topics 6 & 8, topics 4& 3, topics 3& 9 contained similar words and hence are grouped together.

Coherence score for BERT Model

```
#No of topicS =168
bert_model_coherence = CoherenceModel(model=model,texts=processed_docs,dictionary=dictionary,
                                      coherence='c_v')
coherence_bert = bert_model_coherence.get_coherence()
print('\nCoherence Score:',coherence_bert)
```

```
Coherence Score: 1.0
```

Figure 23:Coherence score for BERT Model

4.6 Implementing LDA Model for Topic Modelling

To make a fair comparison of the model performances the LDA model was also utilized for topic modelling from the reviews. Using the same approach as outlined for the BERT model, the data was pre-processed before loading it into the LDA model. The LDA model used a bag of words mechanism by using a vector representation to identify the frequency of words contained in the dataset, before clustering similar words into individual topics.

Creating bag of words

```
[64] bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
```

```
bow_corpus[10:20]
```

```
[[(30, 1), (53, 1), (54, 1), (55, 1), (56, 1), (57, 1), (58, 1)],
 [(43, 1), (44, 1), (49, 1), (59, 1)],
 [(60, 1)],
 [(61, 1)],
 [(43, 1), (62, 1), (63, 1), (64, 1), (65, 1), (66, 1), (67, 1), (68, 1)],
 [(43, 1), (44, 1), (69, 2), (70, 1)],
 [],
 [(71, 1), (72, 1)],
 [],
 []]
```

Figure 24:Creating Bag of Words for LDA Model

4.6.1 Training and Evaluating the LDA Model

Whilst the BERT topic model was able to cluster and identify the total number of topics contained in the review before training the model, the LDA model required specifying the number of topics to be identified by the model. The bag of words was used to extract and identify the latent topics contained in the review.

Figure 25:Topic identification by LDA Model

Upon evaluation, the LDA model had a Coherence score of 0.44 which shows great disparity between the extracted topics by the model and the words contained in the review as contextually.

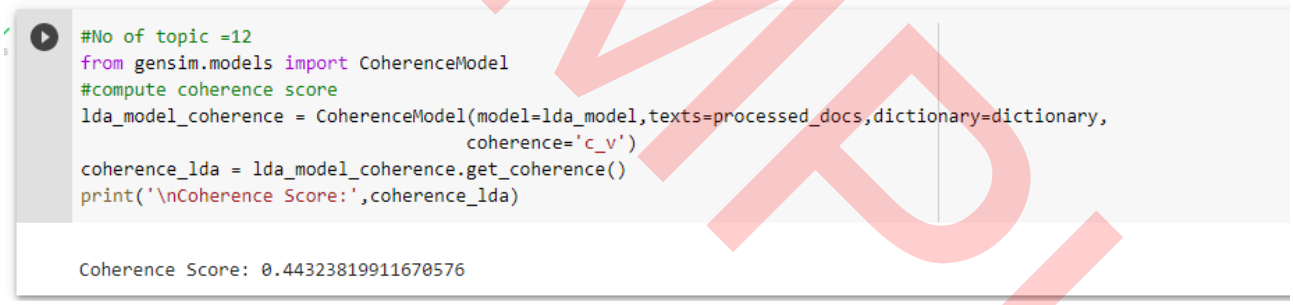


Figure 26: Evaluating LDA Model performance

Figure 27: Word Cloud showing topics extracted by LDA Model

Calculating the coherence score

```
#No of topic =168
from gensim.models import CoherenceModel
#compute coherence score
lda_model_coherence = CoherenceModel(model=lda_model,texts=processed_docs,dictionary=dictionary,
                                     coherence='c_v')
coherence_lda = lda_model_coherence.get_coherence()
print('\nCoherence Score:',coherence_lda)
```

```
Coherence Score: 0.38923880796804433
```

Figure 28:Re-evaluating LDA Model Performance

4.4.1 Analysing the BERT and LDA model for topic modelling

Comparing the performances of the BERT and LDA models for topic modelling, the BERT model automatically identified 168 distinct topics from the data after modelling, while the LDA model required setting the no of topics to identified before modelling. Setting the no of topics to 12 for the LDA model, the model achieved a coherence score of 0.44. However, when the no of topics was increased to 168, to give a fair comparison with the results of the BERT model which had a CV of 1.0, the LDA model's CV dropped to 0.39.

Model	**No of extracted Topics**	**Coherence score**
BERT	168	1.0
LDA	168	0.39

Table 5:Comparing performance of BERT and LDA Models

5.0 Discussion and Findings

This chapter discusses the project's implementation outcomes, comparing them to the original proposed objectives whilst also answering the project's research questions.

The project met its first objective by extracting the apps review dataset from google play store using web scraping techniques. The second objective which in line with the first research question was to identify the most suitable machine learning technique that can be used to analyze sentiment analysis from apps review. This was achieved using the LSTM, SVM, LR, MNB models to which classified the reviews into negative and positive quadrants. In classifying this, the LSTM model performed better with an accuracy of 90.4% accuracy, and 93.47% F1 score. Although the SVM , and Logistic regression model(LR) had higher accuracy score than the LSTM model, the LSTM achieved a better F1 score, which is a better measure of performance in this research due to the unevenness in the data.These results further supports Zhang et al.(2019)'s claim that the LSTM model is better suited for sentiment analysis on apps review with performance better than the other compared models.

The third project objective was to use machine learning technique to extract topics from the reviews and identify the trends within given period. This was achieved using the BERT and LDA model. After training, the BERT model automatically identified 168 distinct topics from the dataset while the LDA model required specifying the number of topics to be identified before training the model. The LDA's model being sensitive to the choice of hyperparameter such no of topics being set before training, posed difficulty in combining the hyperparameter value for optimal performance of the model, whereas, BERT, a pretrained model did not require this.

The CV score of 1.0 achieved by the BERT model upon evaluation indicates that the model accurately identified the underlying topics contained in the data, however, it gives an indication that the model was overfit, which could be largely because of size of data. This further proves to be true Delvin et al., (2018)'s position on the need for using a larger dataset in training BERT models for identifying underlying topics in textual data for optimal performance. Furthermore, compared to the LDA using the bag of words model, which only identified latent topics, the quality of the topics identified by the BERT model provided better context to the themes contained in them. This is because the model was able to capture context and dependencies among words using its self-attention mechanism.

In getting further gain insights from the extracted topics as it relates to the Monzo app functionality, the approach to manually labelling the extracted topics utilizing the visualised word cloud was used. Upon manually labelling the extracted topics, the results showed that for the first 10 ranked topics, 4 of the extracted topics were not descriptive enough to provide any useful information to app developers. The remaining 6 topics in consecutive order related to apps features on account issues, budgeting, account issues, apps ease of use, apps security and upgrade related issues. This hence gives an indication to the top issues being discussed by users of the Monzo app as contained in the utilized dataset. Whilst the manual labelling method provided more insights on the topic discourse, it however relied heavily on knowledge from subject matter experts to achieve this.

Building on the results obtained in this work, it reveals that machine learning is an efficient technique for analysing large data set of apps review by saving up time which is in agreement with existing literatures. Nevertheless, I agree with Gülle et al. (2020) that for the results of machine learning

techniques in extracting topics from text to be useful for organisational use, domain expertise must still be used to manually identify the extracted subjects.

6.0 Conclusion

In concluding this research work, this chapter summarizes the work undertaken in this project to include the adopted research techniques utilized for data collection, pre-processing, model implementation and evaluation. Furthermore, it outlines identified limitations of the project based on the project's outcome and makes recommendations for future works.

The aim and objectives of the project was achieved by identifying sentiments contained in apps reviews and extracting topics from them using machine learning techniques. Based on extensive research conducted reviewing existing literature in apps analysis domain, the LSTM, BERT and LDA models were used with their performances evaluated. The dataset used for this project is the Monzo apps review dataset scrapped from google play store marketplace within a 22-day period which was used for model training and testing.

In evaluating the performance of the models for the outlined tasks, the LSTM model performed best for sentiment analysis with accuracy and F1 score of respectively 90.4% and 93.47%. The BERT model performed better in identifying 168 distinct topics contained in the reviews with the first 6 relevant extracted topics relating to account issues, budgeting, apps ease of use, apps security, upgrade related issues, audio & video features of the app.

Although the project achieved its set out objectives, there were however some limitations encountered while implementing the project discussed below:

1. **The size of the dataset use for topic modelling**: The size of the data set used for training the model was rather small as compared to those utilised in previous similar projects. This was due to time constraint needed to complete the project as apps reviews was collected for a 22-day period. Using a larger dataset of reviews, would have been more ideal to train the machine learning model, providing the model with more words to learn from and to make accurate identification of topics.
2. **Number of models used**: Due to limited number of models used for the project, there was lack of fair basis of comparison on the utilized model's performances as other implemented models might have performed differently in identifying the sentiments in the reviews as well as the underlying topics contained in them.

Topic modelling is undoubtedly a very vast field that is currently receiving attention, with the usage of its methodologies valuable across different industries. This study has utilised its application in the mobile application development domain, however, the approach employed in this research can be employed for topic modelling in other domains. In the future, this project can be improved upon by combining and comparing performances of more machine models in making predictions for sentiment analysis and topic modelling using an all-in-one integrated system. Furthermore, more analysis of the extracted topics can be done by comparing the date that reviews were submitted with their content to identify trends and recurrent patterns.

7.0 References

1. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B. and Kochut, K., 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

2. Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

3. Alessia, D., Ferri, F., Grifoni, P. and Guzzo, T., 2015. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, *125*(3).

4. Asmussen, C.B. and Møller, C., 2019. Smart literature review: a practical topic modelling approach to exploratory literature review. *Journal of Big Data*, *6*(1), pp.1-18.

5. Al Moubayed, Noura, Stephen McGough, and Bashar Awwad Shiekh Hasan. "Beyond the topics: how deep learning can improve the discriminability of probabilistic topic modelling." *PeerJ Computer Science* 6 (2020): e252.

6. A. Ciurumelea, A. Schaufelbühl, S. Panichella and H. C. Gall, "Analyzing reviews and code of mobile apps for better release planning," *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2017, pp. 91-102, doi: 10.1109/SANER.2017.788461

7. Bai, H., Chen, Z., Lyu, M.R., King, I. and Xu, Z., 2018, October. Neural relational topic models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 27-36).

8. Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), pp.993-1022.

9. Cao, Z., Li, S., Liu, Y., Li, W. and Ji, H., 2015, February. A novel neural topic model and its supervised extension. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29, No. 1).

10. C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," 2013 10th Working Conference on Mining Software Repositories (MSR), 2013, pp. 41-44, doi: 10.1109/MSR.2013.6624001.

11. CHEN, Ning; LIN, Jialiu; HOI, Steven C. H.; XIAO, Xiaokui; and ZHANG, Boshen. AR-Miner: Mining informative reviews for developers from mobile app marketplace. (2014). *ICSE 2014: 36th International Conference on Software Engineering: Proceedings: May 31-June 7, Hyderabad, India*. 767-778. Research Collection School Of Computing and Information Systems.

12. Chen, D., & Liu, B. (2015). Recurrent neural networks for sentiment analysis of short texts. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1462-1467.

13. Chen, D., & Liu, B. (2016). A hybrid deep learning model for sentiment analysis of short texts. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2223-2228

14. Ciurumelea, Adelina, et al. "Analyzing reviews and code of mobile apps for better release planning." 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2017.

15. Deng, L. and Yu, D., 2014. Deep learning: methods and applications. *Foundations and trends® in signal processing*, *7*(3–4), pp.197-387.

16. Deveaud, R., SanJuan, E. and Bellot, P., 2014. Accurate and effective latent concept modelling for ad hoc information retrieval. *Document numérique*, *17*(1), pp.61-84.

17. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

18. DHAR, S. and I. BOSE, 2022. Walking on air or hopping mad? Understanding the impact of emotions, sentiments and reactions on ratings in online customer reviews of mobile apps. *Decision Support Systems*, 162, 113769–

19. Gan, Z., He, X., Loy, C.C., and Smola, A. (2016). Top2Vec: Learning Topical Word Embeddings from Text. In Proceedings of the International Conference on Machine Learning (ICML).

20. Gao, Cuiyun, et al. "Ar-tracker: Track the dynamics of mobile apps via user review mining." *2015 IEEE Symposium on Service-Oriented System Engineering*. IEEE, 2015.

21. Guzman, E., Maalej, W., 2014. How do users like this feature? A fine grained sentiment analysis of app reviews.. In: Gorschek, T., Lutz, R.R. (Eds.), RE. IEEE Computer Society, pp. 153–162. URL: http://dblp.uni-trier.de/db/conf/re/re2014. html#GuzmanM14

22. Haroon Malik, Elhadi M. Shakshuki, Wook-Sung Yoo,Comparing mobile apps by identifying 'Hot' features,Future Generation Computer Systems,Volume 107,2020,Pages 659-669,ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.02.008.

23. He, Y., & Wang, S. (2020). Sentiment analysis of app reviews using GPT. In 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 1-6. IEEE.

24. H. Khalid, E. Shihab, M. Nagappan and A. E. Hassan, "What Do Mobile App Users Complain About?," in *IEEE Software*, vol. 32, no. 3, pp. 70-77, May-June 2015, doi: 10.1109/MS.2014.50.

25. Huiliang Zhao, Zhenghong Liu, Xuemei Yao, Qin Yang, A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach, Information Processing & Management, Volume 58, Issue 5,2021, 102656, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2021.102656. (https://www.sciencedirect.com/science/article/pii/S0306457321001448)

26. H. Wu, W. Deng, X. Niu and C. Nie, "Identifying Key Features from App User Reviews," *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 922-932, doi: 10.1109/ICSE43902.2021.00088.

27. Islam, M.S., Hasan, M.M., Wang, X., Germack, H.D. and Noor-E-Alam, M., 2018, May. A systematic review on healthcare analytics: application and theoretical perspective of data mining. In *Healthcare* (Vol. 6, No. 2, p. 54). MDPI.

28. JHA, N. and A. MAHMOUD, 2019. Mining non-functional requirements from App store reviews. Empirical software engineering : an international journal, 24(6), 3659–3695

29. 1. Martens, D., Maalej, W. Towards understanding and detecting fake reviews in app stores. Empir Software Eng 24, 3316–3355 (2019). https://doi.org/10.1007/s10664-019-09706-9

30. Joshi, M., Avesani, P., & Moschitti, A. (2016). Deep learning for sentiment analysis: A survey. arXiv preprint arXiv:1606.04ERC.

31. Kaluarachchi, T., Reis, A. and Nanayakkara, S., 2021. A review of recent deep learning approaches in human-centered machine learning. *Sensors*, *21*(7), p.2514.

32. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

33. K. J. Gülle, N. Ford, P. Ebel, F. Brokhausen and A. Vogelsang, "Topic Modeling on User Stories using Word Mover's Distance," 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2020, pp. 52-60, doi: 10.1109/AIRE51212.2020.00015.

34. K. A. Neuendorf. The Content Analysis Guidebook. Sage Publications, 2002

35. Krishnakumari, K.; Sivasankar, E.; Radhakrishnan, S. Hyperparameter tuning in convolutional neural networks for domain adaptation in sentiment classification (HTCNN-DASC). *Soft Comput.* **2020**, *24*, 3511–3527

36. Lee M, Lee H, Kim Y, Kim J, Cho M, Jang J, Jang H. Mobile App-Based Health Promotion Programs: A Systematic Review of the Literature. Int J Environ Res Public Health. 2018 Dec 13;15(12):2838. doi: 10.3390/ijerph15122838. PMID: 30551555; PMCID: PMC6313530.

37. Liang, T.P., Li, X., Yang, C.T. and Wang, M., 2015. What in consumer reviews affects the sales of mobile apps: A multifacet sentiment analysis approach. *International Journal of Electronic Commerce*, *20*(2), pp.236-260.

38. Li Deng and Dong Yu (2014), "Deep Learning: Methods and Applications", Foundations and Trends® in Signal Processing: Vol. 7: No. 3–4, pp 197-387. http://dx.doi.org/10.1561/2000000039

39. LI, Y. *et al.*, 2017. Mining User Reviews for Mobile App Comparisons. *Proceedings of ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(3), 1–15

40. Li, H., Li, X., & Li, Y. (2018). Sentiment analysis of app reviews using hybrid deep learning model. In 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 1-6. IEEE.

41. Liang, Ting-Peng, et al. "What in consumer reviews affects the sales of mobile apps: A multifacet sentiment analysis approach." *International Journal of Electronic Commerce* 20.2 (2015): 236-260.

42. Luiz, Washington, et al. "A feature-oriented sentiment rating for mobile app reviews." *Proceedings of the 2018 World Wide Web Conference*. 2018.

43. LIU, Y. *et al.*, 2018. Analyzing reviews guided by App descriptions for the software development and evolution. *Journal of software : evolution and process*, 30(12), e2112–n/a

44. Liu, B., Chen, L., & Ma, H. (2018). Sentiment analysis of mobile app reviews using convolutional neural networks and long short-term memory. In Proceedings of the 2018 ACM International Conference on Management of Data (SIGMOD), pp. 635-640

45. Maalej, W., Kurtanović, Z., Nabil, H. et al. On the automatic classification of app reviews. Requirements Eng 21, 311–331 (2016). https://doi.org/10.1007/s00766-016-0251-9

46. Martens, D., Maalej, W. Towards understanding and detecting fake reviews in app stores. Empir Software Eng 24, 3316–3355 (2019). https://doi.org/10.1007/s10664-019-09706-9

47. Nicholas, Jennifer, et al. "The reviews are in: a qualitative content analysis of consumer perspectives on apps for bipolar disorder." *Journal of medical Internet research* 19.4 (2017): e7273.

48. Necmiye Genc-Nayebi, Alain Abran,A systematic literature review: Opinion mining studies from mobile app store user reviews, Journal of Systems and Software,Volume 125,2017,Pages 207-219, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2016.11.027.(https://www.sciencedirect.com/science/article/pii/S0164121216302291)

49. O. Oyebode, F. Alqahtani and R. Orji, "Using Machine Learning and Thematic Analysis Methods to Evaluate Mental Health Apps Based on User Reviews," in *IEEE Access*, vol. 8, pp. 111141-111158, 2020, doi: 10.1109/ACCESS.2020.3002176.

50. Palani, Sarojadevi, Prabhu Rajagopal, and Sidharth Pancholi. "T-BERT--Model for Sentiment Analysis of Micro-blogs Integrating Topic Model and BERT." *arXiv preprint arXiv:2106.01097* (2021).

51. Pagano, D. and Maalej, W., 2013, July. User feedback in the appstore: An empirical study. In *2013 21st IEEE international requirements engineering conference (RE)* (pp. 125-134). IEEE.

52. Panichella, Sebastiano, et al. "How can i improve my app? classifying user reviews for software maintenance and evolution." *2015 IEEE international conference on software maintenance and evolution (ICSME)*. IEEE, 2015.

53. Qi, Jiayin, et al. "Mining customer requirements from online reviews: A product improvement perspective." *Information & Management* 53.8 (2016): 951-963.

54. Suciati, A.; Budi, I. Aspect-Based Sentiment Analysis and Emotion Detection for Code-Mixed Review. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 179–186

55. S. Naseem, T. Mahmood, M. Asif, J. Rashid, M. Umair and M. Shah, "Survey on Sentiment Analysis of User Reviews," *2021 International Conference on Innovative Computing (ICIC)*, 2021, pp. 1-6, doi: 10.1109/ICIC53490.2021.969302

56. S. Mohammad, K. Kiritchenko, S. Dorr, and R. Zaidan. (2013). "Crowdsourcing a Word-Emotion Association Lexicon." Computational Intelligence, 29(3), 436-465.

57. TAM, C., SANTOS, D. and OLIVEIRA, T., 2020. Exploring the influential factors of continuance intention to use mobile Apps: Extending the expectation confirmation model. Information Systems Frontiers, 22(1), pp. 243-257.

58. Taylor, D. G., Voelker, T. A. & Pentina, I. (2011). Mobile application adoption by young adults: A social network perspective. International Journal of Mobile Marketing, 6 (2), 60-70.

59. Triantafyllou, Ioannis, Ioannis C. Drivas, and Georgios Giannakopoulos. "How to Utilize my App Reviews? A Novel Topics Extraction Machine Learning Schema for Strategic Business Purposes." *Entropy* 22.11 (2020): 1310.

60. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017

61. Wang, F., Wu, Y. and Qiu, L., 2012, December. Exploiting discourse relations for sentiment analysis. In *Proceedings of COLING 2012: Posters* (pp. 1311-1320).

62. Wang, H., Liu, Y., & Chen, S. (2016). Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 606-615.

63. Yadav, A. and Vishwakarma, D.K., 2020. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, *53*(6), pp.4335-4385.

64. Yang, G., Wen, D., Chen, N.S. and Sutinen, E., 2015. A novel contextual topic model for multi-document summarization. *Expert Systems with Applications*, *42*(3), pp.1340-1352.

65. Zhang, W.J., Yang, G., Lin, Y., Ji, C. and Gupta, M.M., 2018, June. On definition of deep learning. In *2018 World automation congress (WAC)* (pp. 1-5). IEEE.

66. Zhang, Y., Lu, J., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. ACM Computing Surveys, 51(6), 1-42.

67. Zhang, Y., Liu, Y., & Chen, S. (2019). Aspect -level sentiment analysis of app reviews using attention-based LSTM. In Proceedings of the 2019 ACM International Conference on Information and Knowledge Management (CIKM), pp. 943-952.

68. Zhang, Z., & Zeng, D. (2017). Sentiment analysis of mobile app reviews using deep learning. In 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 1-6. IEEE.

69. Z. Xu, G.L. Frankwick, E. Ramirez, Effects of big data analytics and traditional marketing analytics on new product success: a knowledge fusion perspective, J. Bus. Res. 69 (5) (2015) 1562–1566.

70. ZEČEVIĆ, M. *et al.*, 2021. User Perspectives of Diet-Tracking Apps: Reviews Content Analysis and Topic Modeling. *Journal of medical Internet research*, 23(4), e25160–e25160

71. ZHOU, Y. *et al.*, 2021. User Review-Based Change File Localization for Mobile Applications. *IEEE transactions on software engineering*, 47(12), 2755–2770

72. Federal Reserve (2019). Consumers and Mobile Financial Services 2019. Retrieved from https://www.federalreserve.gov/publications/consumers-and-mobile-financial-services-2019.pdf

73. Source: "The Benefits of Analyzing App Reviews for App Development," Apptentive, 2021

APPENDICES

Ethical clearance for research and innovation projects

Project status

Status

Approved

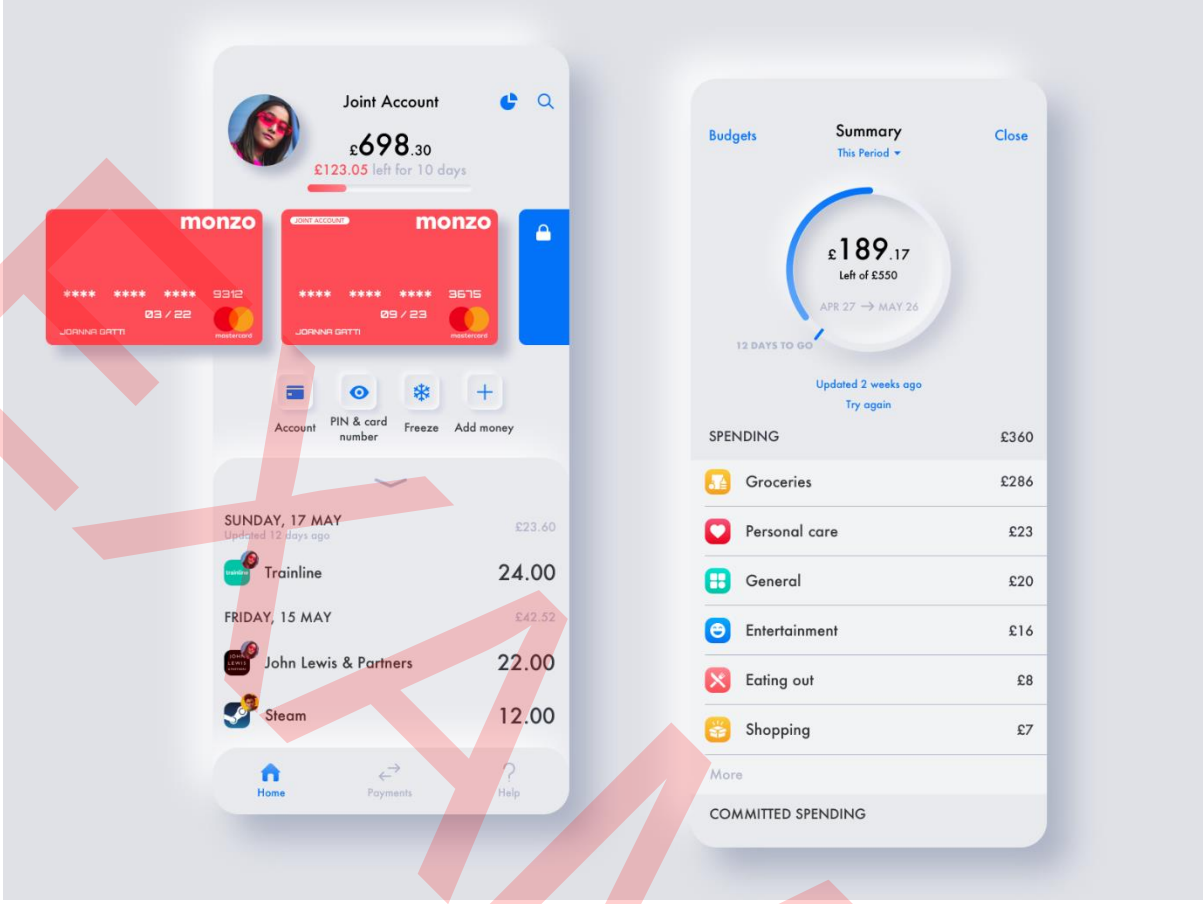
Actions

Date	Who	Action	Comments

Monzo APP review dataset

	A	B	C	D	E	F	G
1	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreat
2	129510a0-b042-45f4-9f0a-3c497014652d	Jose Teles	https://play-lh.googleusercontent.(	The only problem is when you cash out ,cash machine charg	3	0	5.5.0
3	1616dc35-a879-43c6-bc7b-183388e4cde2	Manchu_uk	https://play-lh.googleusercontent.(	It takes age's to receive card's, i have applied 10 dats ago an	1	0	5.4.0
4	d5f8ad28-78ae-446f-85d1-f6c1a2d1f0ef	Patient Onyealusi	https://play-lh.googleusercontent.(	I've not gotten my Account Number	3	0	5.5.0
5	e219a36b-cad0-4eed-bce3-6687587d0efc	mutungi marvin	https://play-lh.googleusercontent.(	I was denied an account with no explanation whatsoever! I	1	0	5.5.0
6	ad2dd2fd-67fc-4423-bf94-15a2f98cff86	M Absar Asif	Q1tS5hQEay8ta21A	Not a good experience with the app. And also the sign up pı	1	0	
7	02a55dc2-0254-44a9-8459-ef39aef40d6d	Chris Browne	https://play-lh.googleusercontent.(	Easy to navigate. Transfer money and statements. Any issue	5	0	5.5.0
8	e0028a04-3a1a-4ee8-b4ad-7b7960124722	Madhushi Chanch	https://play-lh.googleusercontent.(	Excellent service ðŸ‘	5	0	5.3.0
9	05d5158b-10d4-4ac1-bfdb-5a9487178d69	Monday Akpovet	https://play-lh.googleusercontent.(	excellent customer service and very helpful to manage you	5	0	5.5.0
10	c914c02b-cd80-402f-b43e-81bd145fdc2e	kevin sharp	https://play-lh.googleusercontent.(	great app banking	5	0	5.5.0
11	bdb2a73a-6ec0-45f7-8fe6-c4a2d1b3cd44	Kevin Sperrin	https://play-lh.googleusercontent.(	fantastic. better than a high street bank ,	5	0	4.51.0
12	e9a3e3cf-e8ea-4664-ba8f-34fccff94f41	Jeremiah Maina	https://play-lh.googleusercontent.(	Monzo is my first ever online banking and I'm impressed. It	5	0	5.5.0
13	bf77c551-62b3-4d3d-88a9-9150e0b7827d	Ina Mech	https://play-lh.googleusercontent.(	well i love the app i just wish customer service was a bit bet	5	0	5.5.0
14	48bfa5e4-3e09-4848-8c7a-010357e12b5f	Asamoah Derrick	https://play-lh.googleusercontent.(	thank you	3	0	5.4.0
15	4368f17b-9142-4929-aeba-ed7c91f6c0f1	Abdul Basit Chau	https://play-lh.googleusercontent.(	Satisfy with MonzoðŸ‘	4	0	5.5.0
16	395d0f8a-2fa7-4102-ae81-6924b837f0c4	Max Wheeler	https://play-lh.googleusercontent.(	first bank account I've ever had, but honestly everything is s	5	0	5.4.0
17	84b7c6c7-1359-411a-99ce-2056447056b6	Barra O'Loingsigh	https://play-lh.googleusercontent.(	top class bank top class app 5 star customer service	5	0	5.5.0
18	a666f6ae-918b-4a70-aa19-1e1aa378a84f	Mateusz M	https://play-lh.googleusercontent.(	best bank!!!	5	0	5.5.0
19	bf05a3a0-e93f-41a6-85d6-d46e7bfd8355	Migle U.	https://play-lh.googleusercontent.(	The update haiku absolutely slapped 13/10	5	0	5.5.0
20	3df0dae7-3a13-484e-9594-2d056c402567	Simon Stevenson	https://play-lh.googleusercontent.(	They ask for ID they you ain't even got	1	0	5.5.0
21	c2d94e1d-3310-485f-96e8-05b0b19ea0de	santhosh kumar	https://play-lh.googleusercontent.(	Good one	5	0	5.5.0
22	a2f91d25-6c05-4b3a-93b1-9b2fef101ad3	my tunes Morayo	https://play-lh.googleusercontent.(	I can't find the home icon. It's just fixed on the card icon and	1	0	
23	14ccffcf-ce9f-42fa-aa55-4bc56efd9079	Evan Ramazan	https://play-lh.googleusercontent.(	Wow	5	0	5.4.0
24	9378ce7c-46e5-4a08-969c-35eed0b813c6	Armands Sakovics	https://play-lh.googleusercontent.(	the best bank	5	0	4.51.0
25	ad659a37-6923-418a-a042-89d23f92b3a6	Gublalai Dad	https://play-lh.googleusercontent.(	My monzo card initially didnt arrive by the time stated on m	5	0	5.4.0

Monzo App User Interface



PROJECT START		7/11/2022
TASK	START	END
MILESTONE 1: PILOT STUDY		
Project Proposal	7/11/2022	9/11/2022
Formulate Research Question	10/11/2022	12/11/2022
Create Pilot study Report	13/11/2022	24/11/2022
Pilot Study Submission	25/11/2022	25/11/2022
MILESTONE 2: PROJECT DISSERTATION		
Literature Review	26/11/2022	12/12/2022
MILESTONE 3: PROJECT IMPLEMENTATION		
Data Collection	22/11/2022	5/12/2022
Data Pre-processing	6/12/2022	12/12/2022
Data Visualization	6/12/2022	12/12/2022
Model training and Evaluation	13/12/2022	2/1/2023
MILESTONE 4: PROJECT ANALYSIS		
Result Analysis	2/1/2023	4/1/2023
Findings	2/1/2023	4/1/2023
Conclusions	5/1/2023	6/1/2023
MILESTONE 5: SOFTWARE ARTEFACT		
GUI Development	7/1/2023	10/1/2023
Testing	7/1/2023	10/1/2023
MILESTONE 6: REPORT CONTENT AGGREGATION		
First Draft	11/1/2023	15/1/2023
Second Draft	16/1/2023	17/1/2023
Final Draft	18/1/2023	19/1/2023
MILESTONE 7: PROJECT SUBMISSION		
Artefact Submission	19/1/2023	19/1/2023
Report Submission	20/1/2023	20/1/2023

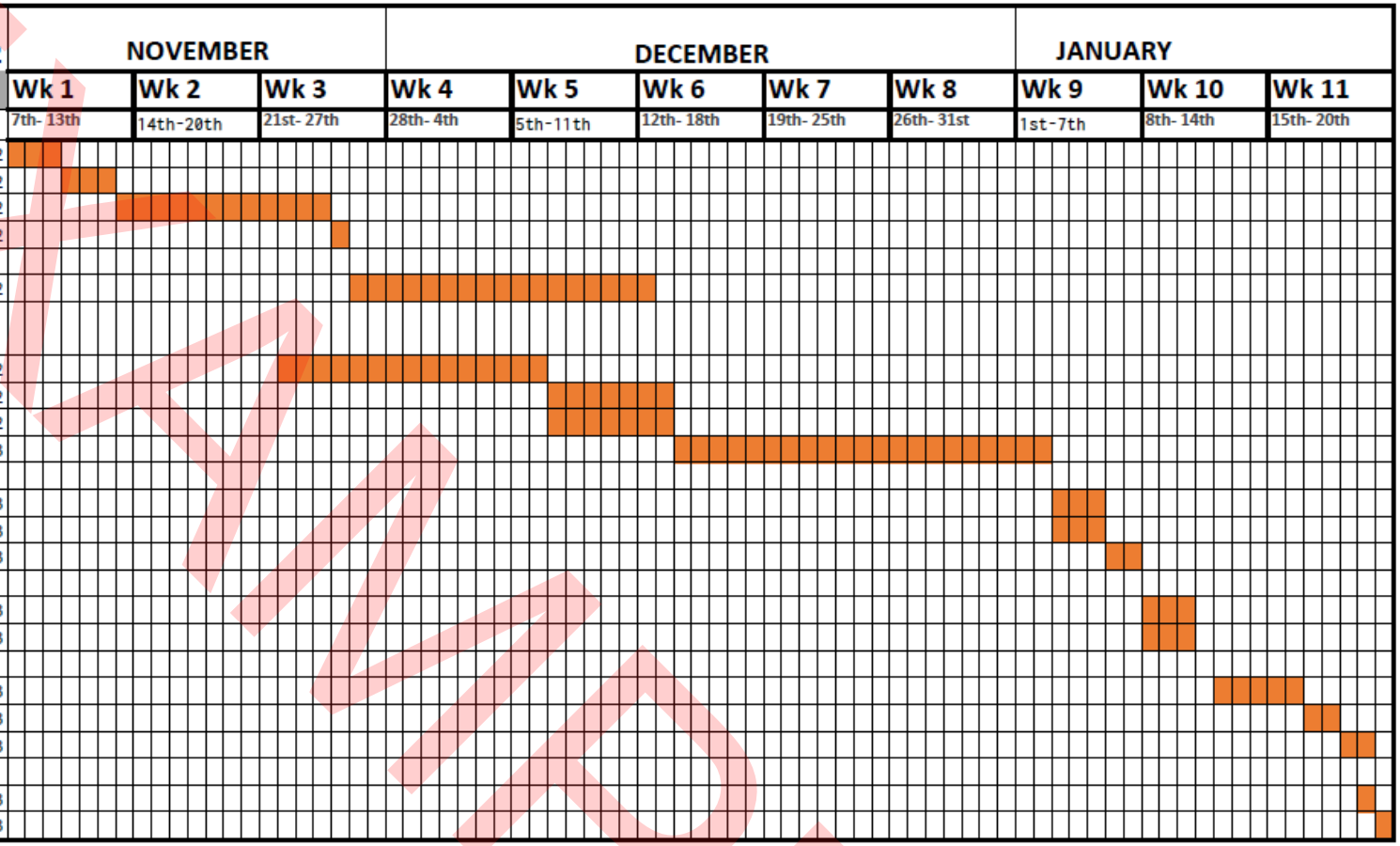


Figure 29:Project Plan