SINHALA CLICKBAIT YOUTUBE VIDEO DETECTION BASED ON THE THUMBNAIL TEXT USING MACHINE LEARNING

A.W.A.T Dilhan

2024

Sinhala Clickbait YouTube Video Detection Based on the Thumbnail Text Using Machine Learning

A.W.A.Tharindu Dilhan

2024

Sinhala Clickbait YouTube Video Detection Based on the Thumbnail Text Using Machine Learning

A dissertation submitted for the Degree of Master of Computer Science

A.W.A. Tharindu Dilhan
University of Colombo School of Computing
2024

Declaration

Name of the student: A.W.A.Tharindu Dilhan
Registration number: 2019/MCS/022
Name of the Degree Programme: Master of Computer Science
Project/Thesis title: Sinhala Clickbait YouTube Video Detection Based on the Thumbnail Text Using Machine Learning

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.

3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.

4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.

5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.

6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
Tharindu	2024/09/20

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	H N D Thilini	Dr. Thepul Ginige	
Signature			
Date	29/09/2024	29/09/2024	

I would like to dedicate this thesis to my supervisor, Dr. H.N.D.Thilini, whose unwavering support, guidance, and expertise have been instrumental in shaping this research, and also to my parents, whose encouragement, and sacrifices have been the cornerstone of my education.

ACKNOWLEDGEMENTS

I extend my deepest gratitude to my supervisor, Dr. H.N.D. Thilini, Senior Lecturer at the University of Colombo School of Computing (UCSC), for giving me the opportunity to complete research.

Her continuous support and invaluable had been pivotal in shaping this work. Dr. H.N.D Thilini's insightful comments, constructive feedback, and profound understanding of the research domain have directly contributed to the success of my research.

I also wish to acknowledge the guidance provided by our MCS Project Coordinator and my co-supervisor, Dr. Thepul Ginige, former head of the National Institute of Business Management, Galle Branch. Their expertise and encouragement have played a significant role in steering me through the intricacies of this research.

Lastly, I express heartfelt appreciation to my parents, and friends, whose unwavering support and encouragement have sustained me through the highs and lows of the research process. Their presence has been a constant source of strength, and this work stands as a testament to their enduring support and encouragement. Thank you, from the bottom of my heart.

ABSTRACT

The YouTube is one of the largest video sharing platforms in the world. There is a mechanism associated with the YouTube platform to earn money by displaying advertisements while playing the YouTube videos. Here, the revenue of the person who posted the video depends on the number of views of the video. Therefore, these videos include intriguing thumbnail with some captivating text to get the user's attention to increase the number of views in order to increase the revenue. Because of that, some people tend to include clickbait statements on YouTube video thumbnails, and those statements are purposely designed to attract the user’s attention and make them curious to follow the link and read, view, or listen to the attached content. It typically employs exaggeration, sensationalism, or curiosity-driven language to attract user's attention. In this research study, There are three main text feature extraction techniques have been employed including countvectorizer, TFIDF vectorizer and Word2Vec word embedding to identify such kind of clickbait content from the thumbnail of a YouTube video and employed different machine learning algorithms including Logistic Regression, Support Vector Machine, Multinomial Naive Bayes and K-Nearest Neighbors with different ranges of N-grams. According to the observed result, Logistic Regression outperformed with the F1 score of 0.81 with the N-Gram range (1,2) and (1,3) along with the TFIDF Vectorization technique.

LIST OF PUBLICATIONS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In today's world online platforms have become powerful tools for the dissemination of information, entertainment, and knowledge. The YouTube, is one of the largest video-sharing platforms for millions of users in the world, and these videos include intriguing thumbnail with some captivating text to get the user's attention to increase the number of views in order to increase the revenue of the person who posted the video.

Because of that, some people tend to include clickbait statements on YouTube video thumbnails which typically leads to exaggeration, sensationalism, or curiosity to attract user's attention to follow the link and view.

This situation is affecting detrimentally on the society. The users are tricked into clicking on those videos and, finally they waste their valuable time and money.

In this research, the challenging problem is to detect the clickbait Sinhala YouTube videos based on the thumbnail text using Machine learning and Optical Character Recognition.

This research delves into the intersection of machine learning, natural language processing, and optical character recognition on the Sinhala language. Here, the specific focus is on the challenging task of extracting and identifying clickbait Sinhala statements on YouTube video thumbnails. This study stems from the critical need to foster a more responsible, inclusive, and respectful online environment. The clickbait content can have far-reaching consequences, impacting not only individual well-being but also societal harmony. By addressing this issue within the Sinhala YouTube community, I contribute to the broader effort of promoting responsible and safe digital environment.

In the context of the Sinhala language processing, there were three main approaches which were Lexical Based Approaches, Machine Learning Approaches and Hybrid Approaches (Ruwandika and Weerasinghe, 2018).

The first sentiment analysis work in Sinhala also was the lexicon-based approach and it has been done in 2015. The developed lexicon was tested against 2083 Sinhala news articles. The Naive Bayes, SVM, and Decision tree algorithms were used in the process of sentiment classification and achieved the accuracy of 60% (Medagoda, 2016).

In 2019, there was a rule-based approach for binary sentiment classification of Sinhala news comments. In that study, a sentiment lexicon was created in a semi-automated way, and, used it for sentiment analysis. According to the obtained results, the Naïve Bayes approach has given the best result (Chathuranga et al., 2019), as well as the first hate speech detection research study in Sinhala language has been done in 2019, and for that both lexicon-based approaches and machine learning based approaches were used. The Naive Bayes classifier achieved the highest accuracy which is 92.3% and the recall value of 84% (Sandaruwan et al., 2019).

Because of the lack of the performance in lexicon-based approaches the usage of the lexicon-based approaches was reduced over the time, and there was a research study that focused on an ensemble approach which combines both lexicon-based approach and the machine learning based approach, to achieve a more accurate result than the individual approaches. That research work pointed out that ensemble method is more accurate than the usage of the individual approaches (Jayasuriya et al., 2020).

As related research, in the context of the Sinhala language processing, the traditional machine learning techniques such as Naïve Bayes, Support Vector Machine have been applied to identify abusive Sinhala comments in social media. In that study, the Multinomial Naïve Bayes (MNB) classifier has achieved the highest accuracy, which is 96.5% (Sandaruwan et al., 2020).

When it comes to the deep learning with respect to the sentiment analysis in Sinhala language. There was a research study which applied both machine learning and the deep learning techniques, the result showed that the RNN with the fast text feature extraction method has the greatest AUC ROC 0.71 with 70% accuracy (Fernando et al., 2022).

Apart from the Sinhala language processing, the major concern in this research study is text content extraction on images.

When it comes to the content extraction on images, even though deep learning algorithms are now the most common approach to text extraction, there are a number of alternative approaches, including basic classifiers using manual features and complex systems that combine different algorithms. However, some systems have highly flexible approaches to learn essential information from labelled data (De Silva et al., 2021).

In a recent study, the researchers developed a new method for text extraction from images that was based on stroke width variation. The method first cleans up the image and removes the background. Then, it uses a CNN to extract both the features and text from the image. This method is particularly useful for extracting text from images with complex backgrounds, such as YouTube thumbnails. It was more efficient than traditional methods, which can be slow and computationally expensive (C. P. Chaithanya et al., 2019).

According to a previous research study in the context of detecting text on natural images, there were three basic image processing techniques identified as preprocessing, text localization, classification, and relevant character identification. In that study, researchers have identified various classification algorithms for text detection in natural images, among those algorithms the CNN has achieved the best performance (Nadarajan, 2018).

Another research study has focused on the Optical Character Recognition with Neural Networks to extract the text from images, and that process involved image segmentation, feature extraction, and classification. When the image was noiseless the accuracy was 100%, and for the noisy images the accuracy was 95% (Garg et al., 2018).

There was only one research study focusing on extraction of Sinhala text content from YouTube thumbnails using Convolutional Spiking Neural Networks. In that study, the researchers have used three convolutional layers. The rate base convolutional spiking neural network approximation is used for train the network. When it comes to the results of this study, it has been able to extract the Sinhala text content from YouTube thumbnails with an accuracy of approximately 85% (De Silva et al., 2021).

When it comes to the content moderation on YouTube platform, there was a research study in Indonesia that used optical character recognition (OCR) and face recognition to detect clickbait YouTube videos. There, the three machine models have been created with the Support Vector Machine. The performance of these individual models was not sufficient, hence the final model has been created by combining those three models, and it was able to achieve the accuracy of 0. 968.That was a significant improvement when compared with the performance of each individual model. The deep learning approach has also been used in another research study on the topic of the clickbait video detection. In that research study, an algorithm called “BaitRadar” has been developed by the researchers. That algorithm comprises of 06 inferences models. The models are combined to make the accurate final result, and the average accuracy was 98% (Gamage et al., 2021).

In the context of the Sinhala language, there was a research study that has focused on the hate content detection in YouTube platform. There, the title of the video, description, user comments, thumbnail text, and tags of the videos were used to extract the features with the techniques of count vectorization, TFIDF vectorization and word2vec. In this study, both the lexicon-based approach and the machine learning based approaches were tested out.

According to the evaluation results, all the machine learning approaches has performed well than the lexicon-based approach. As the final result, the logistic regression model achieved the highest accuracy which was 89% (De Saa and Ranathunga, 2020).

In this research study the proposed approach can be outlined as follows.

- Data Collection: Collects a dataset of Sinhala YouTube videos, specifically focusing on the textual content of their thumbnails. This data serves as the foundation for the analysis. In order to extract the Sinhala text from the YouTube thumbnails, the google cloud vision API is used.
- Labeling: To train the machine learning model, the video thumbnail text in the dataset is manually labeled as either clickbait or non-clickbait.
- Feature Extraction: Machine learning models require numerical input, hence the relevant features from the textual content of video thumbnails are extracted using TFIDF vectorizer.
- Model Selection: Extracted features are passed through the different machine learning algorithms find the best algorithm to build a classification model.
- Training and Validation: The dataset is splatted into training and validation sets to ensure the model's performance. This step helps fine-tune the model and assess its accuracy in identifying clickbait content.
- Testing and Evaluation: After training the model, it is tested on new, unseen Sinhala YouTube videos thumbnail text to evaluate its ability to detect clickbait content accurately. Finally, the model's performance is measured using various metrics, such as accuracy, precision, recall, and F1 score.

This research makes a valuable contribution by developing a specialized system for clickbait Sinhala YouTube video detection, which is a novel and relatively unexplored area. The unique challenges posed by the Sinhala language make this endeavor distinctive and most significant for content analysis in the digital age.

1.1 Motivation

The motivation behind conducting this study is to eliminate the clickbait Sinhala content from the YouTube platform and ensure the secure and safe digital communication among the Sinhala community.

1.2 Statement of the problem

The YouTube, is one of the largest video-sharing platforms for millions of users in the world, and these videos include intriguing thumbnail with some captivating text to get the user's attention to increase the number of views in order to increase the revenue of the person who posted the video. Because of that, some people tend to include clickbait statements on YouTube video thumbnails which typically leads to exaggeration, sensationalism, or curiosity to attract user's attention to follow the link and view.

The users are tricked into clicking on those videos and, finally they waste their valuable time and money.

In this research, I am going to address the challenging problem which is to detect the clickbait Sinhala YouTube videos based on the thumbnail text with Machine Learning and Optical Character Recognition.

1.3 Research Aims and Objectives

The primary focus of this research is to develop a system that can identify clickbait videos on YouTube by analyzing the thumbnail text at the time the video is being uploaded.

1.3.1 Aim

The aim of this research is to create a system that can effectively detect clickbait Sinhala language content from YouTube by analyzing the thumbnail text, improving the online experience and safety for the Sinhala community.

1.3.2 Objectives

- Accurate Clickbait Content Detection:
  Develop a model that can accurately detect clickbait text on Sinhala YouTube video thumbnails. The objective is to achieve a high level of precision and recall in identifying clickbait thumbnails, minimizing false positives and false negatives.
- Improve the YouTube User Experience:
  Enhance the YouTube user experience by reducing exposure to clickbait thumbnails. The objective is to help promoting the authenticity and relevance of the content.
- User Protection:
  Protect YouTube users from deceptive or clickbait content by effectively detecting

them. The objective is to reduce the likelihood of users being misled or lured into clicking on videos with sensationalized thumbnails.

- Content Quality Control:
  Contribute to maintaining content quality and integrity on the YouTube platform. The objective is to assist content creators and platform administrators in identifying and flagging videos that employ clickbait content, ensuring a reliable and trustworthy platform.

1.4 Scope

In this research, the aim is to build an application that can classify Sinhala YouTube video thumbnails as clickbait or not by considering the thumbnail texts which is extracted automatically using the google cloud vision API. In order to train a machine learning model, Sinhala YouTube Video thumbnail text are extracted from the selected YouTube channels in Sri Lanka, and manually prepare a dataset with the classification labels as clickbait or not using 1 and 0 respectively. The prepared dataset runs through the different machine learning and deep learning algorithms to build a model that can classify YouTube video thumbnails as clickbait or not with the best possible accuracy and performance.

The features of the application are,

- Realtime Detection: Implementing the trained model to perform real-time clickbait text detection on new, unseen Sinhala YouTube video thumbnails. This enables the model to provide immediate feedback on the likelihood of a thumbnail being clickbait, and not allowing users to upload the video.
- Image Analyzing and Font free Optical Character Recognition: This application can detect and extract Sinhala text accurately from any given thumbnail with the help of the Optical Character Recognition (OCR) technology and it does not depend on the type of the font. That means it is font free OCR.
- High Effectiveness: Utilizing sophisticated machine learning and deep learning models, such as Logistic Regression, Support Vector Machines (SVM), Recurrent Neural Networks (RNNs) to learn the representations and patterns in the text of the clickbait Sinhala YouTube video thumbnails. These models can effectively capture complex relationships and dependencies within the textual data.
- User Interface Integration: Developing a user-friendly interface that allows users to

interact with the clickbait video thumbnail detection system. This includes functionalities to upload thumbnails and receive detection results based on the model's predictions.

The limitations of the application are,

- Limited Dataset: Availability of a limited amount of labeled clickbait Sinhala YouTube video thumbnail data could pose a challenge. Deep learning models often require large amounts of diverse data to achieve optimal performance. Insufficient data may limit the model's ability to generalize well to unseen misleading thumbnails.
- Evolution of Clickbait Strategies: Clickbait strategies employed by content creators continuously evolve to bypass detection systems. New techniques may emerge that the model has not been trained on, leading to potential detection limitations. Regular updates and retraining of the model may be required to keep up with evolving misleading tactics.

1.5 Structure of the Thesis

The chapter 01 of this report provides an overview of the project with the comprehensive understanding of the field of study and the scope.

Chapter 02 of this report includes a detailed discussion of the previous research studies in this field and also describes the current knowledge and new pathways related to the study.

Chapter 03 of this report includes the experimental design of the research including system architecture, tools and technologies used, and also the research output, and its usage to address the said problem.

In chapter 04 of the report includes the detailed illustration of the result and evaluation with respect to the different tools, technologies and approaches that had been used in the experimental design.

In the final chapter of this report has summarized the research work, findings and contribution and also have outlined the directions for future works.

CHAPTER 2

LITERATURE REVIEW

This chapter shall give essential background information referring to published material in research papers, URLs (from credible sources), magazine articles and similar. This chapter should include a critical review of similar research published in recent years in credible publications such as journals and peer reviewed conferences etc.

2.1 A Literature Review

With the rapid development of digital media platforms, user-generated content on platforms like YouTube has grown exponentially. However, this growth has also brought with it an increase of clickbait and inappropriate content. Detecting and mitigating clickbait content, in Sinhala language becomes a significant challenge due to its low resource availability. This literature review focuses on the emerging field of clickbait Sinhala text content detection, specifically on YouTube thumbnails, utilizing machine learning techniques.

2.1.1 Sinhala Language Processing

The enhancement of the machine learning and deep learning techniques has led to improve the performance of the Natural Language Processing tasks on the high resource languages such as English and Chinese. However, the Sinhala language is a language which has low resources, therefore, there was no significant improvements on Sinhala language(Senevirathne et al., 2020). Whereas, there are number of researches have been published on the topics such as sentiment analysis, hate speech detection and clickbait content detection in Sinhala language. In the context of the Sinhala language processing, there were three main approaches which were Lexical Based Approaches, Machine Learning Approaches and Hybrid Approaches(Ruwandika and Weerasinghe, 2018). Lexical based approaches believe that the most important part of the text classification is being able to understand the lexical phrases. In Lexical based approach, the machine was fed with grammar of the language, manually created rules to describe the texts, or the domain knowledge which is used to describe a certain text(Ruwandika and Weerasinghe, 2018).

The first sentiment analysis work in Sinhala also was the lexicon-based approach and it has been done in 2015. In that research, the sentiment lexicon of Sinhala language has been developed using sentiWordNet 3.0 and an online Sinhala/English dictionary(Medagoda et al., 2015). The dictionary contains synonyms for Sinhala words and their English representation. Each adjective and adverb in sentiWordNet were looked up in the developed Sinhala/English dictionary. The sentiment score for the original word was calculated using the score of the Sinhala word and the related synonyms. There were few assumptions. The sense of the word for a given context for both languages is the same. The sentiment score of a word in both languages is the same. When it comes to the sentiment analysis, only the adjectives and adverbs were considered as the most important language units. However, the problem of this approach was, there were no direct meaningful translation for some of the important words that is used to express a sentiment. However, the developed lexicon was tested against 2083 Sinhala news articles. The Naive Bayes, SVM, and Decision tree algorithms were used in the process of sentiment classification and obtained the accuracy value up to 60%. After that, in 2016, using the same dataset, the sentiment classification model has been built with the help of the feedforward Neural Network(Medagoda, 2016). There, they have used TF-IDF and binary values of frequently occurring adjectives and adverbs.

In 2019, there was a rule-based approach for binary sentiment classification of Sinhala news comments. In that approach, a sentiment lexicon was created in a semi-automated way, and, used it for sentiment analysis. The sentiment of a given sentence was determined by aggregating the sentiment score of individual words in a particular sentence which were calculated according to the sentiment lexicon. For the sentiment analysis, Naive Bayes, SVM and decision trees algorithms were used. According to the obtained results, the Naïve Bayes approach has given the best result(Chathuranga et al., 2019).

The first hate speech detection research study in Sinhala has been done in 2019, and for that both lexicon-based approaches and machine learning based approaches were used. In that study the researchers applied several preprocessing steps, including removing non-Sinhala characters, URLs, emojis, stop words, and punctuation, as well as they used shallow stemming. They analyzed various features such as Bag of Words, Word n-grams, and character n-grams, with the machine learning classifiers such as SVM, RF, and NB. Among the features, character tri-grams performed the best, and the Naive Bayes classifier achieved the highest accuracy which is 92.3% and a recall value of 84%. Because of the high morphological nature of the Sinhala language character n-gram feature gave the best results in their research. (Sandaruwan et al., 2019).

However, the lexicon-based approaches are not suitable in sentiment analysis in social media, because of the informal language usage in the social media. As a solution, there was an ensemble approach which combines both lexicon-based approach and the machine learning based approach together. In that research work, the sentiment analysis has been done by considering the comments on sport related YouTube videos. Here, in lexicon-based analysis, a Bayesian analysis method have been used, and a subjective lexicon has been constructed using the training dataset and also it has been expanded using a Sinhala-English dictionary. In the machine learning approach, the classification has been done using the Naive Bayes classifier, Logistic Regression Classifier and support vector machine classifier, and each algorithm was trained using unigram, bigram and trigram feature extraction methods. As the ensemble approach, the machine learning methods and the lexicon-based methods were combined using a majority voting ensemble classifier. In machine learning approach the best results were achieved with the Logistic Regression unigram classifier with stop-word removal among all the other machine learning algorithms with respect to the accuracy and F1-score. For the lexicon-based method the best performance was achieved Levenshtein ratio analysis. When considering the individual methods, the machine learning approaches were more accurate than the lexicon-based approaches. However, this research work point outs that ensemble method is more accurate than the usage of the individual approaches (Jayasuriya et al., 2020). In the context of the Sinhala language there are limited number of research that is based on lexicon approach, and using also the recognized hatred terms will leads to low accuracy(Davidson et al., 2017). Due to the increasing vocabulary and the scaling up issues lexicon based research studies have been decreasing over time(Gamage et al., 2022).

As related research, in the context of the Sinhala language processing, the traditional machine learning techniques such as Naïve Bayes, Support Vector Machine have been applied to identify abusive Sinhala comments in social media. In this research, the researchers have developed three different models which are Multinomial Naïve Bayes, Support Vector Machine, and Random Forest Decision Tree. The features were extracted from Bag of Word model, word n-gram model, character n-gram model, and word skip-gram model to detect Sinhala abusive comments. In this study, the Multinomial Naïve Bayes (MNB) classifier achieved the highest accuracy, which is

96.5%, and also an average recall of 96% for both character tri-gram and character four-gram models. (Sandaruwan et al., 2020). However, when it comes to the machine learning in the context of the natural language processing, it considers only the present or absent of the particular words in the dataset to generate a result. Hence, the machine learning models do not capture the information about word's meaning or context. Whereas when it comes to the deep learning with the techniques such as word embedding, it vectorizes words as multi-dimensional continuous floating-point numbers, and syntactically and semantically similar words are mapped to the proximate points in a geometric space. Hence, it captures the syntactically and semantically similar words in the text. Therefore, the deep learning approaches performs well on the natural language processing domain. Whereas deep learning models requires large amount of data to obtain the accurate results. Therefore, in some of the use cases machine learning has performed well than the deep learning approaches, or there may be a similar probability to happen in other way round as well. According to the results of a published research in the context of the sentiment analysis in Sinhala language, performed using both machine learning and the deep learning techniques, the deep learning has outperformed machine learning. The result of that research study has shown that the RNN with the fast text feature extraction method has the greatest AUC ROC 0.71 with 70% accuracy apart from the other deep learning and machine learning approaches(Fernando et al., 2022).

2.1.2 Content Extraction from Images

There are many research studies have been done in the context of text content extract extraction from images. In this section the aim is to discuss the limitations and challenges of the existing research studies. Even if the extraction of the text from an image became a major concern in many researches, extraction of Sinhala text from a YouTube thumbnail becomes a challenging task because of the various background objects are associated with the text of the thumbnail image. Even though, mainly deep learning algorithms are used for extracting the texts from images there are alternative approaches, including from the basic classifiers using manual features to complex multistage systems that combine different algorithms for text detection. Key features used in these methods include edge features, shape contexts, and texture descriptors. Apart from those, some systems have highly flexible approaches to learn essential information from labelled data(De Silva et al., 2021). In a recent research study, focusing on stroke width in images, the researchers have employed a filtering technique for image preprocessing to improve image quality and conducted segmentation to isolate the region of interest, eliminating the background. After that, the non-text region is removed upon finding the maximally stable extremal regions. Here, the stroke width was calculated using the stroke width variation. The Convolutional Neural Networks were used to extract both the features and text from the data (C. P. Chaithanya et al., 2019). According to a previous research study which was to detect text on natural image, there were three basic image processing techniques identified as preprocessing, text localization, classification, and relevant character identification. In that study, researchers have identified various classification algorithms for text detection in natural images, including Support Vector Machine, Adaboost, and CNNs. Among those algorithms the CNN has performed better for detecting text from images than others(Nadarajan, 2018).

The detection, recognition and prediction of orientation of the text in an indoor image has been addressed with respect to the Urdu language in a recent research study. As the initial step, they used a custom algorithm called FasterRCNN with some CNNs. Then, they used a custom network (RRNN) to guess the ligature's orientation. Finally, they used a Two-Stream Deep Neural Network (TSDNN) to identify the ligatures(Arafat and Iqbal, 2020). Another research study has focused on the Optical Character Recognition with Neural Networks to extract the text from images, and that process involves image segmentation, feature extraction, and classification. When the image was noiseless the accuracy was 100%, and for the noisy images the accuracy was 95%(Garg et al., 2018). A research study has been done for detecting, localizing, and recognizing text in natural scene images. It includes three parts, in the preprocessing stage, they create a text region detector using the Histogram of Oriented Gradients (HOG). Then, they segment connected components using local binarization, and they distinguish between text and non-text components based on their normalized height-width ratio and compactness. The text recognition has been done with a distance metric feature extraction approach which was based on zone centroid and image centroid(Pise and Ruikar, 2014).

However, even if there were numbers of research studies had been conducted on the context of the text content extraction from images, none of them performs well for the text content extraction on YouTube thumbnails with the significant level of accuracy because of the complexity of the image background, and also there was only one research study focusing on extraction of Sinhala text content from YouTube thumbnails using Convolutional Spiking Neural Networks. In that study, the researchers have used three convolutional layers. The rate base convolutional spiking neural network approximation is used for train the network, and there are three main steps in the proposed solution which are pre-processing, prediction and postprocessing. The preprocessing step is responsible for removing unnecessary background noise and colors from the thumbnail, identifying text regions and segmenting the text regions in to words and characters. The post-processing step is responsible for took the predicted results and developed the text by concatenating the characters, words, and sentences as appears in the thumbnail. When it comes to the results of this study, it has been able to extract the Sinhala text content from YouTube thumbnails with an accuracy of approximately 85%(De Silva et al., 2021).

2.1.3 Content Moderation on YouTube Platform

YouTube is one of the largest video sharing platforms in the world. It allows users to create their own videos and upload. Even if YouTube maintains their community guidelines for content moderation, including the removal of hate speech and harassment, these guidelines often rely on user reporting.

Hence, some users upload clickbait content to YouTube in order to increase the number of views of the videos, because it leads to increase their revenue. When it comes to the clickbait YouTube videos, they always contain a thumbnail with some narrative or clickbait text and images which gives incentive to watch it. In order to eliminate this kind of YouTube videos from YouTube platform a number of research studies have been done in different contexts.

There was a research study in Indonesia that used optical character recognition (OCR) and face recognition to detect clickbait YouTube videos based on the content of the thumbnails. Here, the videos that pitted Muslim preachers on YouTube were considered as the clickbait video. The researchers have paid attention to identify the faces of people in thumbnails while detecting the text content on the thumbnail. Because, the primary contribution of that study is to demonstrate that optical character recognition and facial recognition can effectively address the clickbait YouTube thumbnail detection. For that, the titles and the YouTube video thumbnails have been collected and labeled as “Prov” and “Nonprov”. In that study, there were four main stages called OCR preprocessing, OCR processing, text alteration, and model training and testing. OCR preprocessing involves image processing and adding new fonts to improve the OCR engine to read text on images. OCR processing collects all text from thumbnails using the OCR engine and stores it for further processing. Text alteration includes number conversion, special character processing, lowercase conversion, root word searching, names and foreign terms searching. Finally, the text alteration data were used to create the word vectors and those word vectors had been used to train the machine learning models. Here, the three machine models have been created with the Support Vector Machine, Model 1 with Optical Character Recognition and it produces an accuracy value of 0.732. Model 2 with facial recognition and it produces an accuracy of 0.8. Model 3 with text alteration and it produces an accuracy of 0.808. The performance of these individual models was not sufficient, hence the final model has been created by combining those three models, and produced an accuracy of 0.968, a sensitivity of 0.968, a precision value of 0.9698, and an F1-Score of 0.9678(Vitadhani et al., 2021).

Another research study on the topic of the clickbait video detection has been done with the deep learning. Here, the researchers have developed an algorithm called "BaitRadar". This algorithm uses six inferences models, and these models are individually trained on the attributes of the video including title, comments, thumbnail, tags, video statistics and audio transcript. Here, the text-based models utilize the long short-term memory (LSTM) networks, and the thumbnail model utilizes a convolution neural network (CNN) to extract the features in the thumbnail. The models are combined and computed the average and experimented with transfer learning to make the accurate final result even though some input attributes are unavailable. This approach was tested on 1,400 YouTube videos, and the average accuracy was 98%(Gamage et al., 2021).

A study has been done in the context of the fake news classification using multi model dataset that contains both the text and image data from the famous YouTube channels. The purpose of this study was to build a, multi-modal dataset of fake news collected from YouTube platform for researchers to use. The data has been labeled into 2-way and 6-way classes based on categories of fake news such as misleading content, manipulated content, satire or parody, and false connection according to text, thumbnail images, and content provided in videos. Additionally, different transfer learning models are also used to classify fake news. Transfer learning techniques have been applied to the dataset, using BERT for text-only data, and Resnet50, Resnet151, Inception, VGG-16, and VGG-19 for image-only data. For the multi-modal data, the fusion techniques have been applied using distil-BERT and VGG-16 to predict labels, and the accuracy was 73%. Using the base-BERT and Inception models for the 2-way classification the highest accuracy which was 91% achieved(Fatima et al., 2023).

In the context of the Sinhala language, a research study has been done for the hate content detection in YouTube platform. This approach is associated with the title, description, user comments, thumbnail text, and tags of the videos. In this case, the features were created by analyzing the thoughts and feelings expressed in the YouTube data. In order to extract the features, count vectorization, TFIDF vectorization and word2vec have been used. Bag of words, word vectors were extracted using Counter vectorization, TFIDF vectorization was used to extract words based on importance. To extract features based on semantic similarity, the word2vec CBOW text vectorization method was used. The extracted features were used to build the different classification models namely Logistic Regression, Multinomial Naïve Bayes, Random Forest Classifier and Artificial neural network. In this study a lexicon-based approach was also used to identify the hate speech. However, according to the evaluation results, all the machine learning approaches were able to achieve accuracies greater than 70%. Lexicon based approach got the lowest accuracy which was 52%. Moreover, The logistic regression model achieved the highest accuracy which was 89% among all the classifiers(De Saa and Ranathunga, 2020).

However, above all research studies are talking about the content moderation on YouTube platform with respect to the published content on the YouTube platform such as title of the video, description, video statistics and the content of the thumbnail, and also one study mentioned the consideration of the audio transcript of the video to detect the misleading content. Usage of the audio transcript is truly leads to an accurate result, but it is not applicable for all the use cases, because in some of the videos, audio quality is poor and also the generated transcripts will not be accurate. Another study above, mentioned that the usage of face recognition to detect videos that pitted Muslim preachers on YouTube in Indonesia. The usage of the face recognition was a great deviation from the regular approaches to detect the clickbait content on YouTube platform.

However, with the consideration of the above research studies, the need for a new research study is shown to prevent clickbait Sinhala videos from being added to YouTube. The proposed approach is truly different from all the studies above mentioned, because the videos are filtered out at the time they upload, by analyzing the thumbnail text.

CHAPTER 3

METHODOLOGY

3.1 Research Philosophy

The main intention of this research is to develop a machine learning model focusing on the thumbnail text that can accurately distinguish Sinhala YouTube Videos that are likely to be clickbait and those that are not. This study aligns with the pragmatist philosophy in the context of recognizing the need for a practical and context-specific approach to understand and address the issues of clickbait Sinhala YouTube videos. Pragmatism allows us to combine the elements of positivism and interpretivism which acquire the importance of both objective measurements and contextual understanding in the development of an effective solution.

3.2 Research Design

The design of this research study is based on both qualitative and quantitative methods to provide a holistic view of the research problem. This design allows for a more comprehensive exploration of clickbait text content in Sinhala YouTube videos thumbnails, employing statistical analysis with insights from human experiences.

3.3 Proposed Solution Design and Implementation Details

3.3.1 System Architecture:

The proposed system is based on the development of a machine-learning model for clickbait Sinhala YouTube video detection, focusing on the analysis of thumbnail text. The system employs Natural Language Processing (NLP) techniques to extract relevant features from the text, incorporating machine learning-based approaches to detect clickbait YouTube video thumbnails.

3.3.2 Technologies and Techniques:

Google Cloud Vision API: The cloud vision API is used to extract the Sinhala statements from the video thumbnails.

Natural Language Processing (NLP): NLP techniques are used to analyze the thumbnail text of the YouTube video to identify the clickbait statements.

Machine Learning (ML): Implementing a supervised learning model, including Random Forest, Gradient Boosting, Support Vector Machine, K-Nearest Neighbors, Logistic Regression and Multinomial Naive Bayes to classify the thumbnail text into clickbait and non-clickbait categories.

3.3.3 System Implementation

Entire process of the experiment is divided into 05 major steps as Data collection and annotation, Data preprocessing, Feature extraction, Model selection and training, and Performance evaluation.

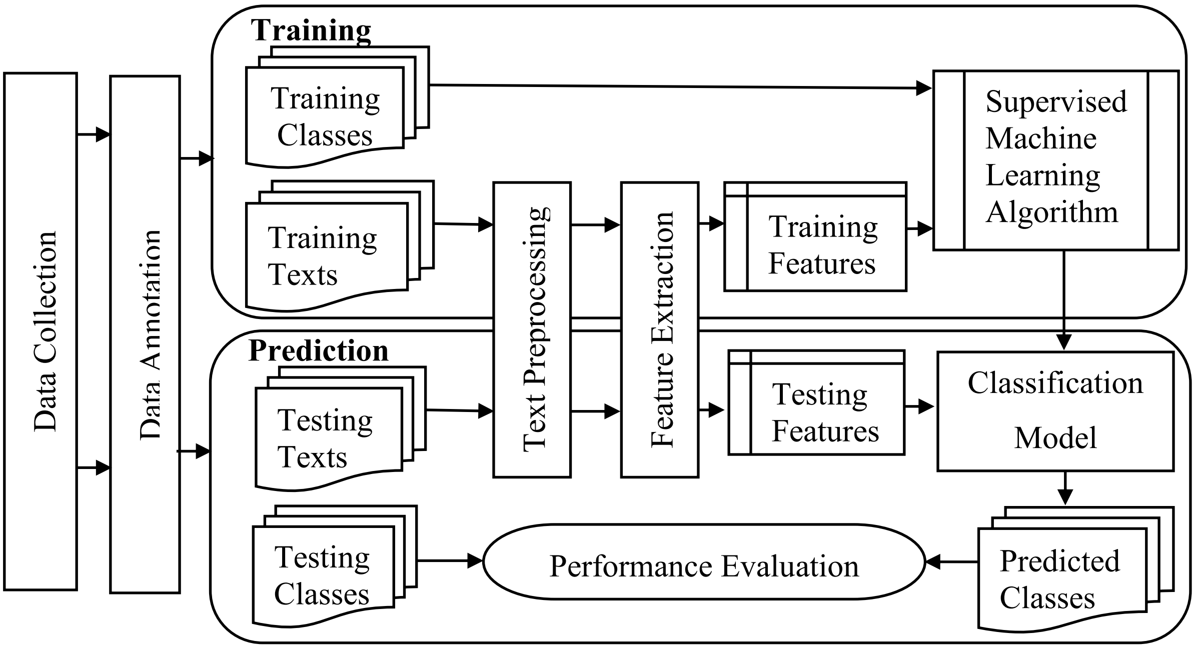


Figure 3.1: Design of the Research

Data Collection: To perform a successful experiment on clickbait Sinhala YouTube video detection based on the thumbnail texts, the availability of a rich dataset is really important. A dataset of Sinhala YouTube video thumbnail texts is collected from multiple Sinhala YouTube channels belong to Sri Lanka. The dataset contains 650 records. Table 1 presents an overview of the number of clickbait YouTube thumbnail texts and none clickbait YouTube thumbnail texts.

Table 3.1: Classes Distribution of the Dataset

Number of Clickbait YouTube Thumbnail texts	325
Number of None clickbait YouTube Thumbnail texts	325

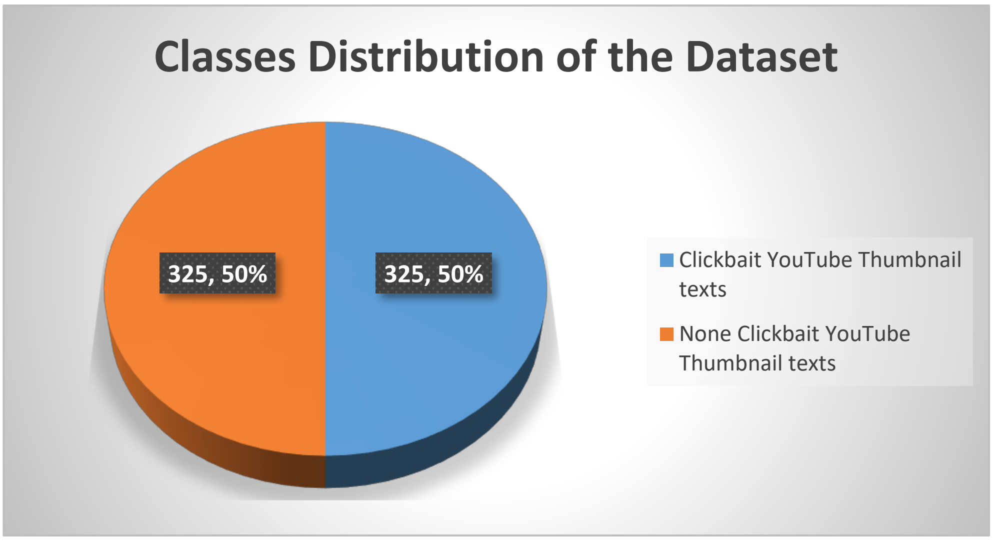
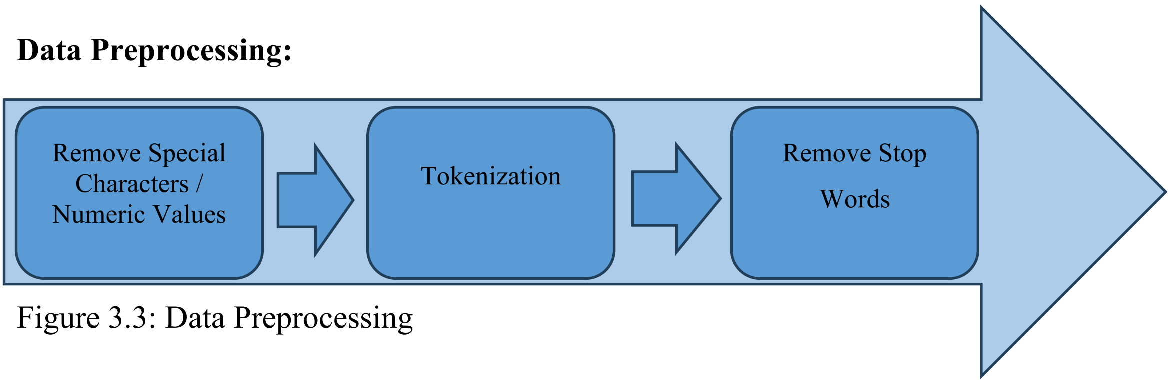


Figure 3.2: Classes Distribution of Dataset

Data Annotation: Each record in the dataset is manually annotated whether is it belongs to clickbait or none clickbait class. content is purposely designed to attract the user's attention and make them curious to follow the link, and it typically employs exaggeration, sensationalism, or curiosity-driven language to attract user's attention. The collected dataset consists of 650 records of data. Among them 325 records have been annotated as the clickbait content and 325 records of data have been annotated as none clickbait content using the label '1' and '0' respectively. 80% from the annotated dataset was considered as the training dataset, and the 20% from the annotated dataset was considered as the testing dataset.

Data Preprocessing:



Remove Special Characters / Numeric Values

Tokenization

Remove Stop Words

Figure 3.3: Data Preprocessing

In order to reduce the noise, the basic data preprocessing steps are applied to the dataset. Initially the special characters, punctuation marks and numeric values of the dataset are removed using a user defined function, and then the tokenization is applied to the dataset using the 'sinling' tokenizer in order to generate meaningful features for the machine learning model. Finally, stop words of the dataset are removed to further enrich the dataset and a data frame is created using the preprocessed data.

Feature extraction: A set of textual features are extracted from the tokenized Sinhala text data, for that both the Count Vectorizer and the Term frequency-inverse document frequency (TF-IDF) vectorizer are used.

- Count Vectorizer

  Count Vectorizer is a tool available in the scikit-learn library in python and it transforms the text data into vectors based on the frequency of each word that occurs in the entire dataset. Each row in the matrix corresponds to a document, and each column corresponds to a unique word in the text dataset. The values in the matrix represents the counts of how many times each word appears in each document.

- TF-IDF Vectorizer

  Term frequency – inverse document frequency vector is a way measure the importance of a particular word or term in the dataset. Using TF-IDF Vectorizer we can find the most important word as a feature in a particular text dataset.

  Term Frequency (TF) measures how often a term appears in a document. It is calculated as the ratio of the number of times a term appears in a document to the total number of words in that document.

$$\boxed{\textit{Term Frequency } TF(t,d) = \frac{\textit{count of } t \textit{ in } d}{\textit{number of words in } d}}$$

Figure 3.4: Function for Term Frequency

Inverse Document Frequency (IDF) measures the importance of a term across the entire dataset. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term.

$$\boxed{\textit{Inverse Document Frequency} = \log_2\left(\frac{\textit{number of documents } N}{\textit{number of documents containing the term } t}\right)}$$

Figure 3.5: Function for Inverse Document Frequency

The TF-IDF score for a term in a document is the product of its Term Frequency and Inverse Document Frequency.

$$\boxed{tf - idf(d,t) = tf(t,d) * idf(t)}$$

Figure 3.6: Function for Term Frequency and Inverse Document Frequency

Model selection: A suitable machine learning model will be selected by applying the dataset to different machine learning algorithms including Logistic Regression, Multinomial Naive Bayes, Random Forest, Support Vector Machine, Gradient Boosting, and K-Nearest Neighbors with different N- Gram levels, and also applying the word2vec word embedding technique to classify the thumbnails into clickbait and non-clickbait categories.

Performance evaluation: During the evaluation process, the model performance will be assessed by employing various performance metrics including accuracy, precision, recall, and F1 score concerning the frequency-based text features extraction techniques and N-Gram levels and also focusing on the word embedding techniques.

Deployment of the Model: The Machine learning model which is going to be considered as the best is deployed as a Streamlit web application which user can select and upload a particular thumbnail, and then the web application notifies the probabilistic percentage to become a clickbait video considering the selected thumbnail.

CHAPTER 4

EVALUATION AND RESULTS

In the result and evaluation section will compare the performance matrices including accuracy, precision, recall, and F1 score of different machine learning models concerning the different text feature extraction techniques and different N – gram levels.

The accuracy refers to the ratio of correctly predicted instances to the total number of instances in the test dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN =False Negatives

Precision refers to the how many instances are actually positive instances out of the total positively predicted instances.

$$Precision\ = \frac{TP}{TP + FP}$$

Recall measures the proportion of true positive predictions among all actual positive instances in the dataset.

$$Precision\ = \frac{TP}{TP + FN}$$

The F1 score is calculated as the harmonic mean of precision and recall. It combines precision and recall into a single value. The value of the F1 score lies between 0 to 1 with 1 being a better.

$$F1\ Score\ = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In this experiment, the text data have been divided in to two sections which is for training and testing with the ratio of 80:20, which means that 80 % for training and 20% for testing.

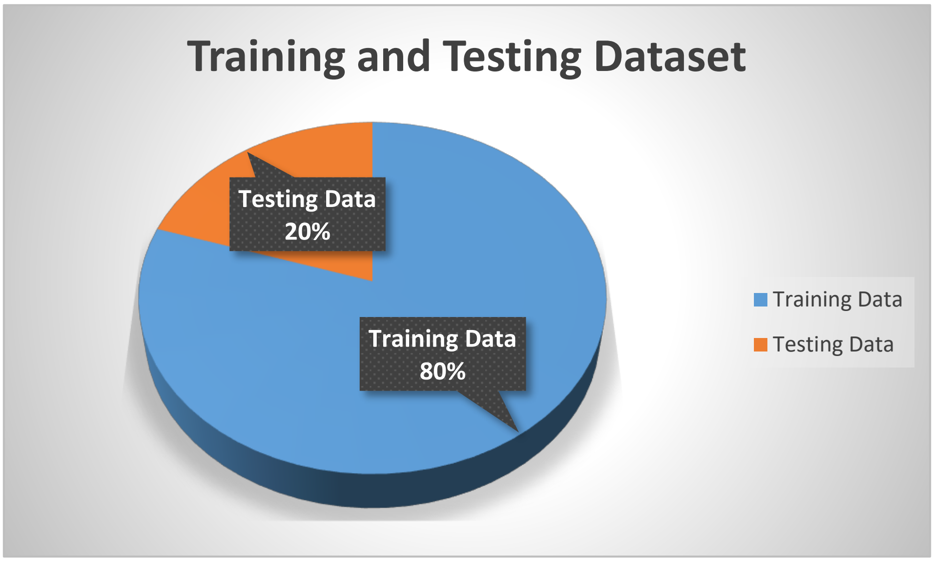


Figure 4.1: Structure of the training and testing dataset

Prepared training dataset contains 520 records, and among those, 260 records were manually annotated as clickbait and 260 records as none clickbait using 1 and 0 respectively.

The features of the text data were extracted using frequency-based techniques including count vectorizer and TFIDF vectorizer incorporating with different N-gram ranges, and also using word embedding technique called word2vec. Then the extracted text features were passed for different machine learning algorithms, and the performance of each machine learning model was evaluated using a test dataset. However, N-grams cannot be applicable for the word2vec word embedding technique because each word is represented as a dense vector in a high-dimensional space, and this representation doesn't directly rely on the sequence of words but rather on the contexts in which words appear.

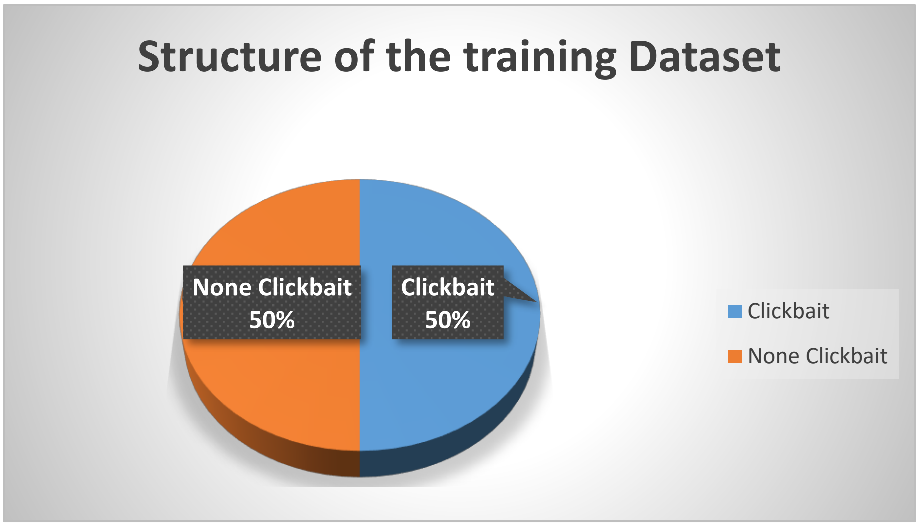
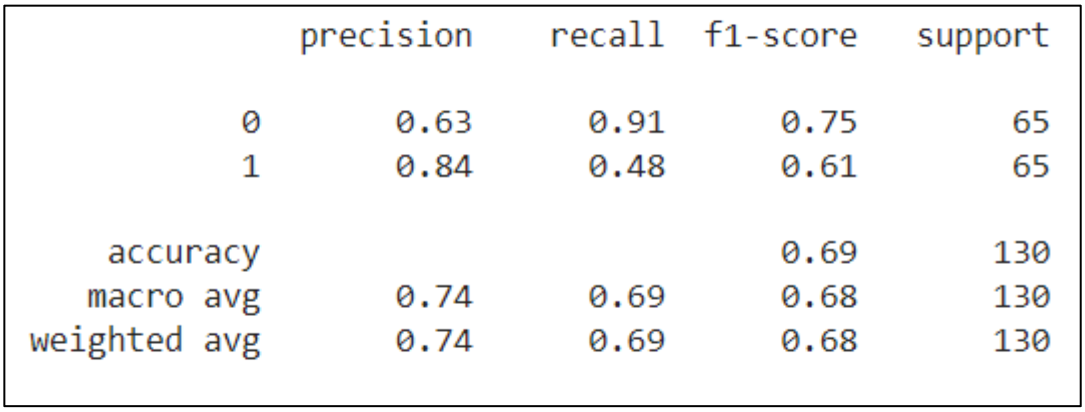


Figure 4.2: Structure of the training Dataset

4.1Countvectorizer

4.1.1 Logistic Regression with N- gram range (1,1)

Figures 4.3 and 4.4 present the classification report and confusion matrix for the Logistic Regression model using count vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.63      0.91      0.75        65
           1       0.84      0.48      0.61        65

    accuracy                           0.69       130
   macro avg       0.74      0.69      0.68       130
weighted avg       0.74      0.69      0.68       130
```

Figure 4.4: Classification Report - Logistic Regression with N - gram range (1,1)

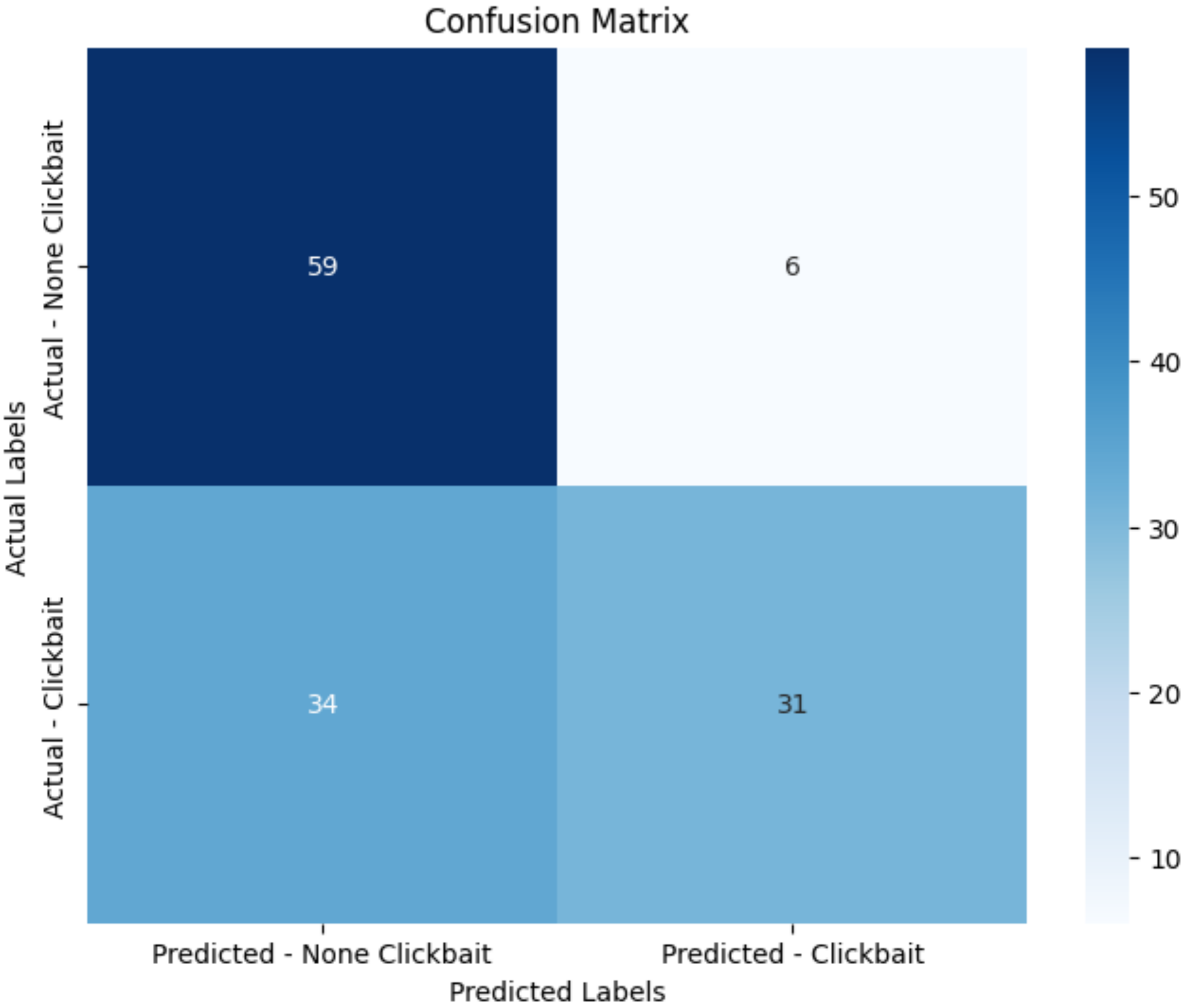
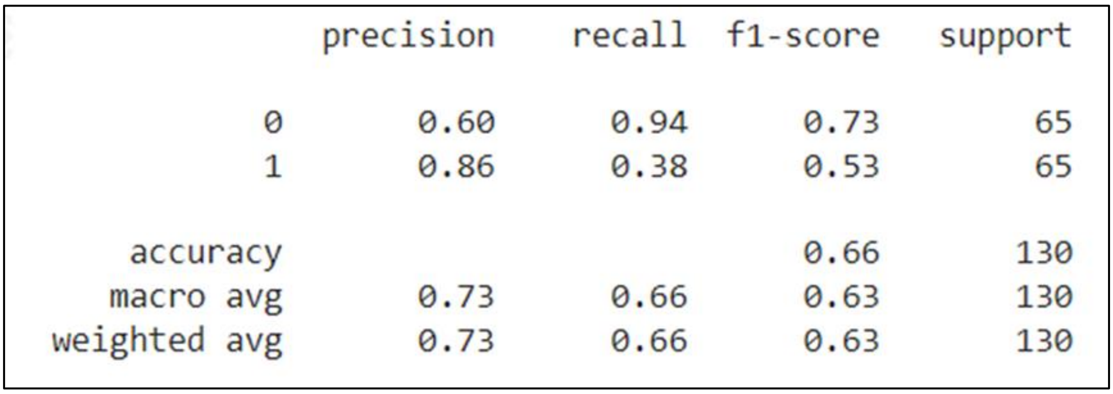


Figure 4.3: Confusion Matrix - Logistic Regression with N - gram range (1,1)

4.1.2 Logistic Regression with N- gram range (1,2)

Figures 4.5 and 4.6 present the classification report and confusion matrix for the Logistic Regression model using count vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.60      0.94      0.73        65
           1       0.86      0.38      0.53        65

    accuracy                           0.66       130
   macro avg       0.73      0.66      0.63       130
weighted avg       0.73      0.66      0.63       130
```

Figure 4.5:Classification Report - Logistic Regression with N - gram range (1,2)

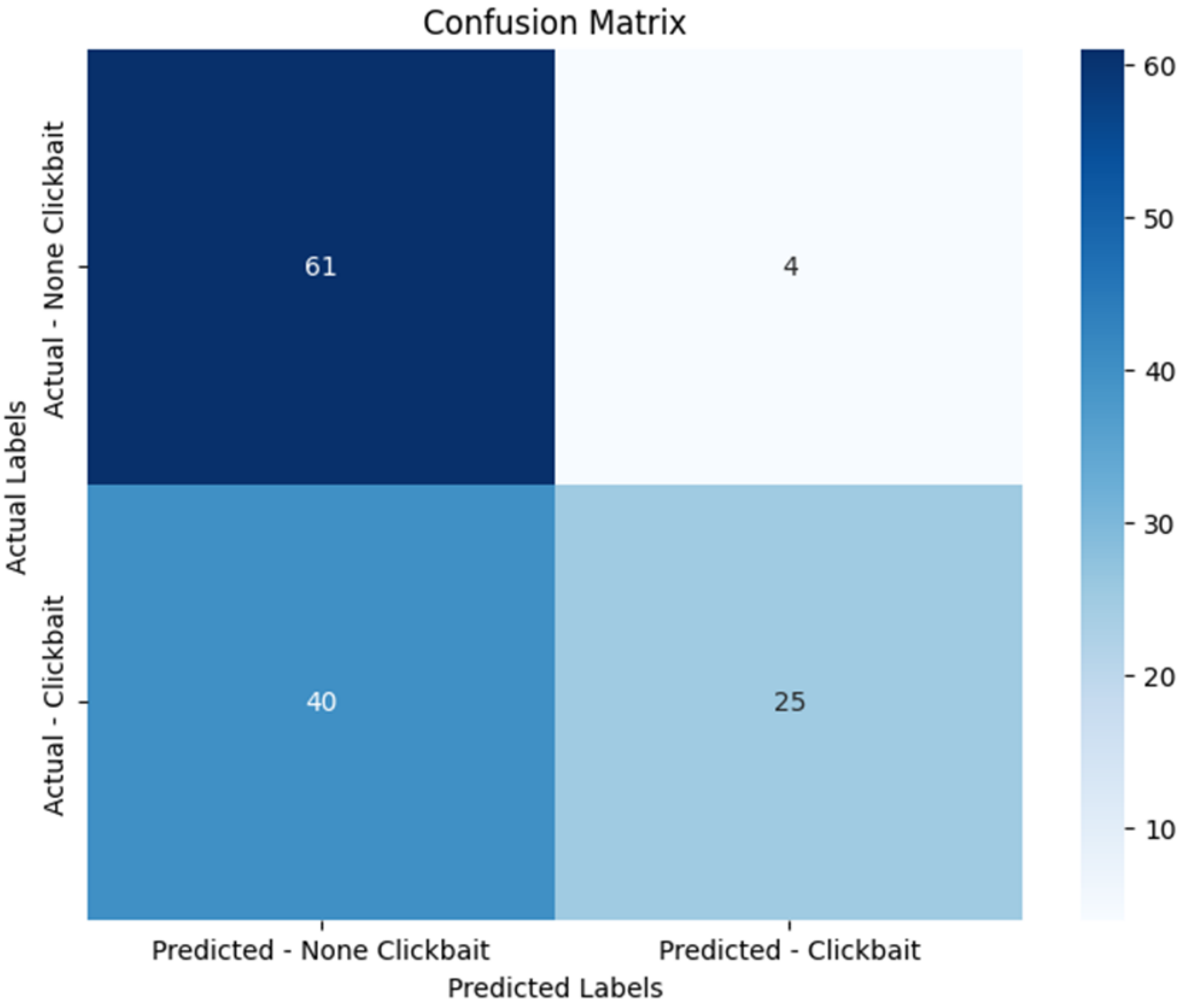
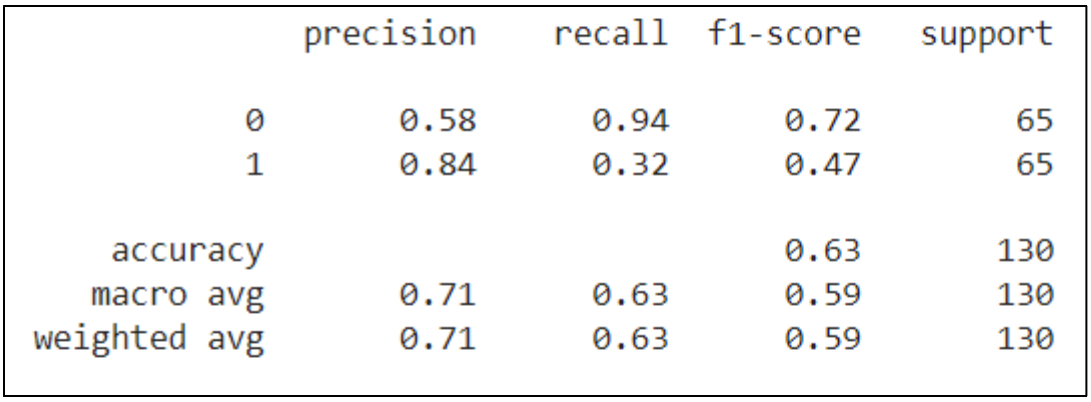


Figure 4.6: Confusion Matrix - Logistic Regression with N - gram range (1,2)

4.1.3 Logistic Regression with N- gram range (1,3)

Figures 4.7 and 4.8 present the classification report and confusion matrix for the Logistic Regression model using count vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.58      0.94      0.72        65
           1       0.84      0.32      0.47        65

    accuracy                           0.63       130
   macro avg       0.71      0.63      0.59       130
weighted avg       0.71      0.63      0.59       130
```

Figure 4.7: Classification Report - Logistic Regression with N- gram range (1,3)

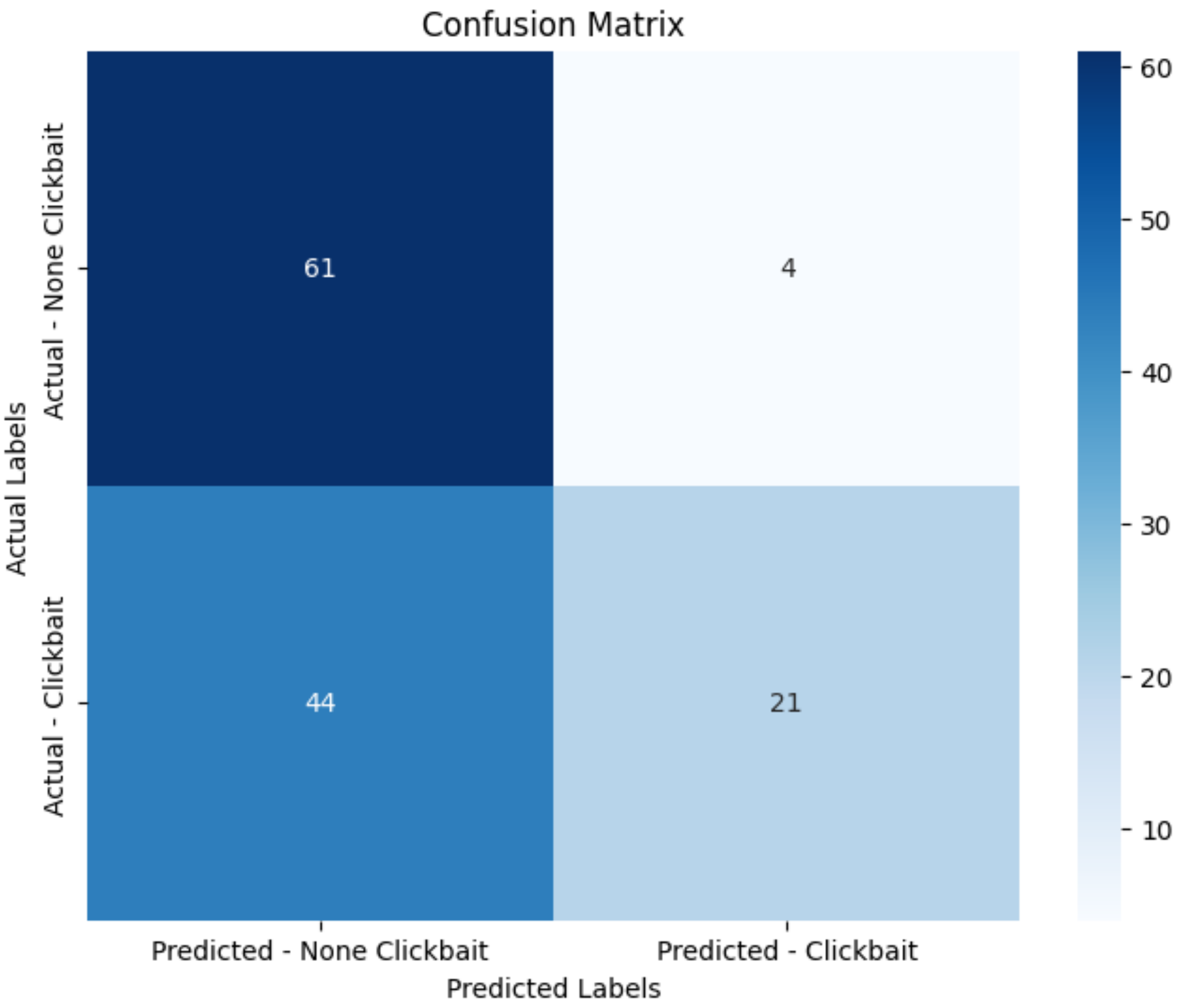
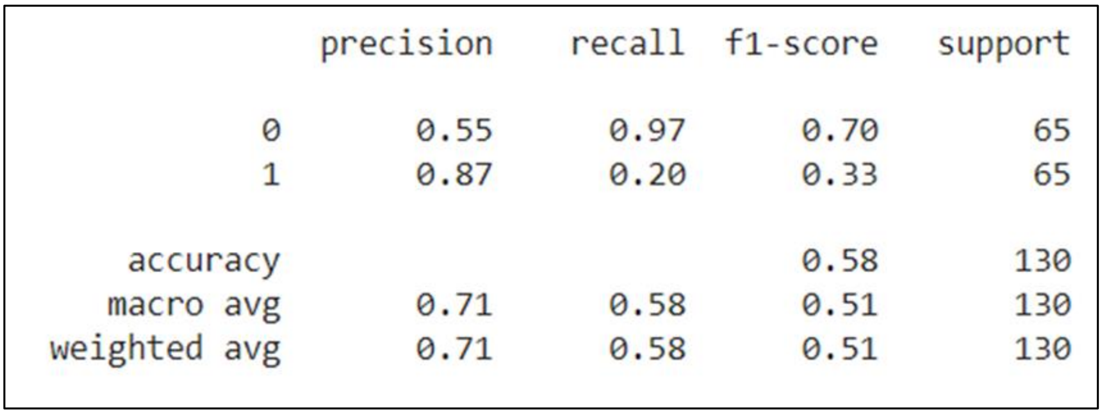


Figure 4.8: Confusion Matrix - Logistic Regression with N - gram range (1,3)

4.1.4 Support Vector Classifier with N- gram range (1,1)

Figures 4.9 and 4.10 present the classification report and confusion matrix for the Support Vector Classifier model using count vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.55      0.97      0.70        65
           1       0.87      0.20      0.33        65

    accuracy                           0.58       130
   macro avg       0.71      0.58      0.51       130
weighted avg       0.71      0.58      0.51       130
```

Figure 4.9:Classification Report - Support Vector Classifier with N- gram range (1,1)

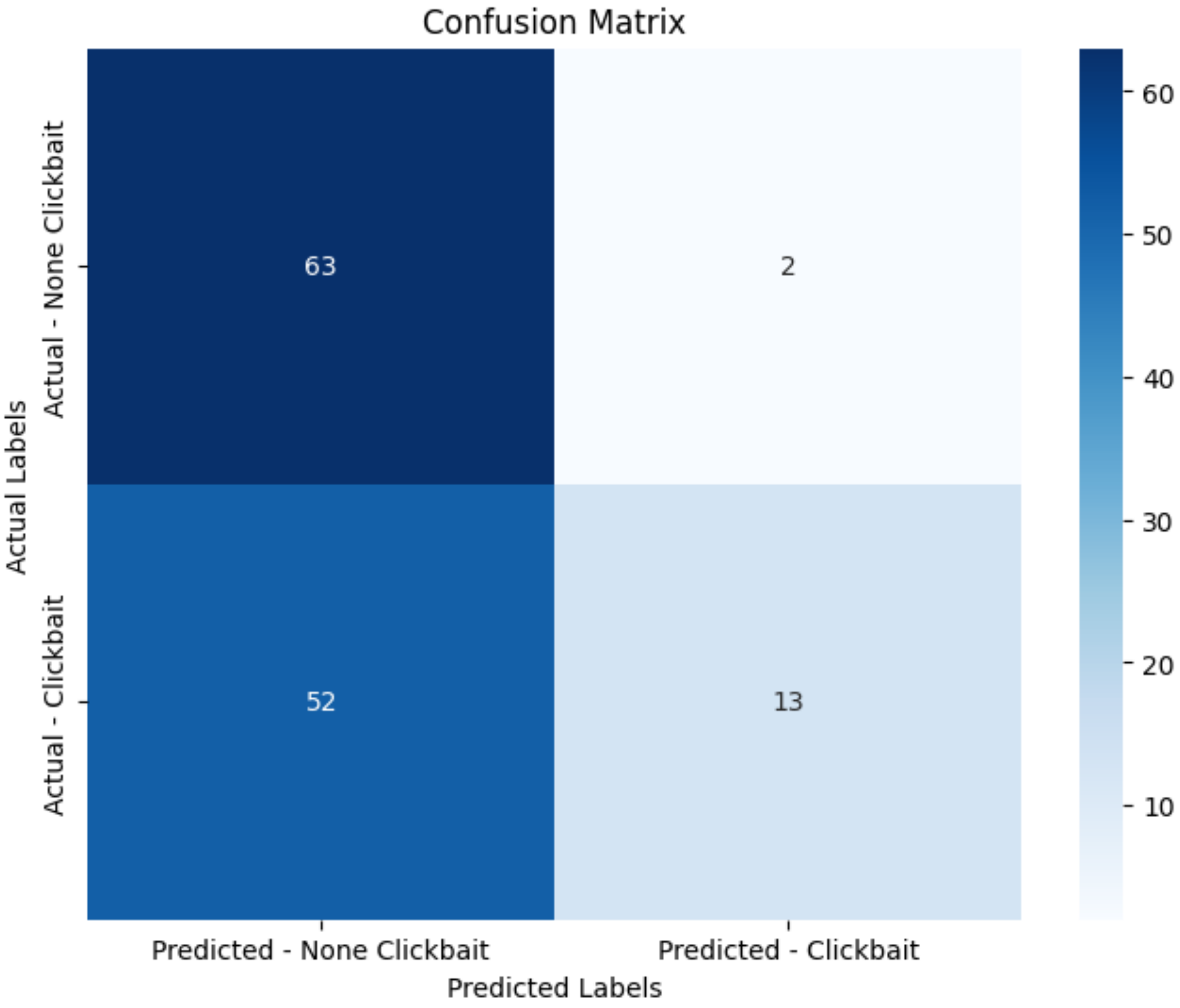
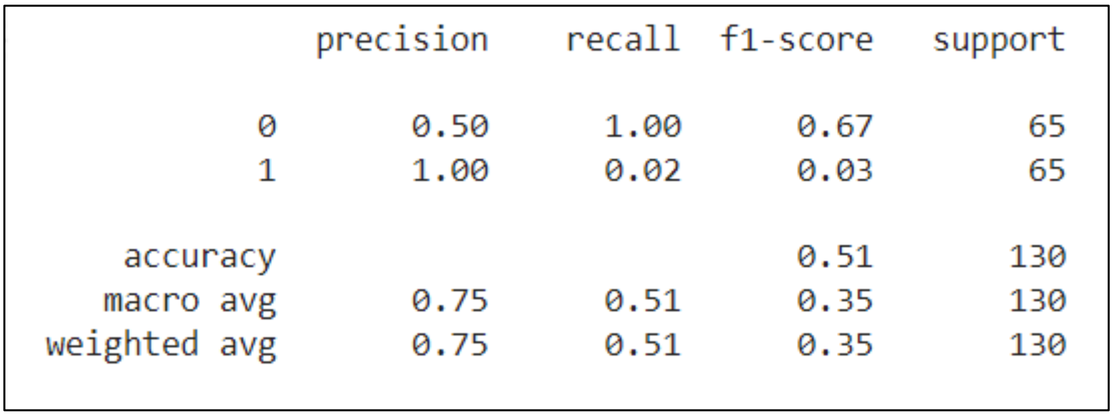


Figure 4.10:Confusion Matrix - Support Vector Classifier with N- gram range (1,1)

4.1.5 Support Vector Classifier with N- gram range (1,2)

Figures 4.11 and 4.12 present the classification report and confusion matrix for the Support Vector Classifier model using count vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        65
           1       1.00      0.02      0.03        65

    accuracy                           0.51       130
   macro avg       0.75      0.51      0.35       130
weighted avg       0.75      0.51      0.35       130
```

Figure 4.11:Classification Report - Support Vector Classifier with N- gram range (1,2)

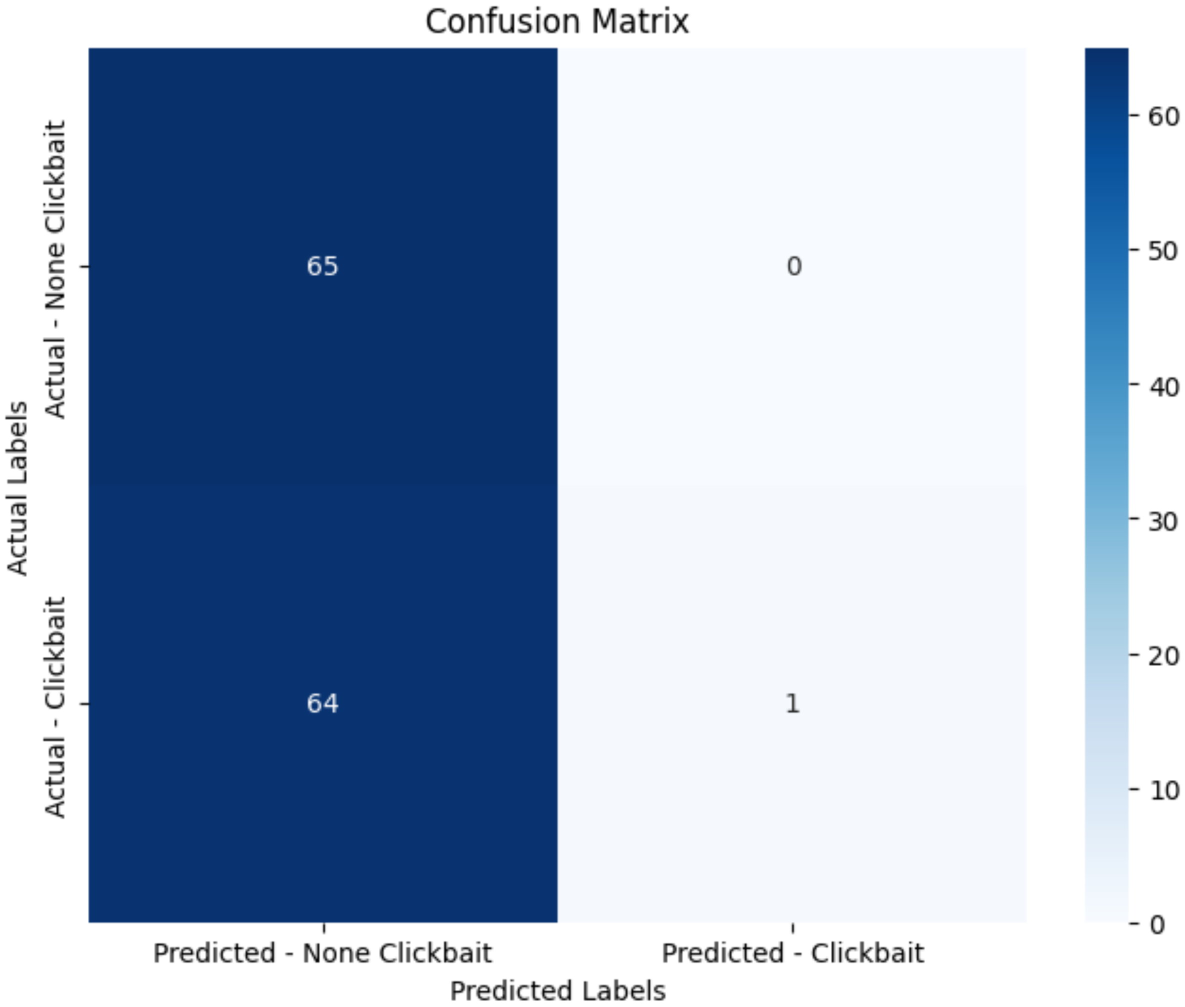
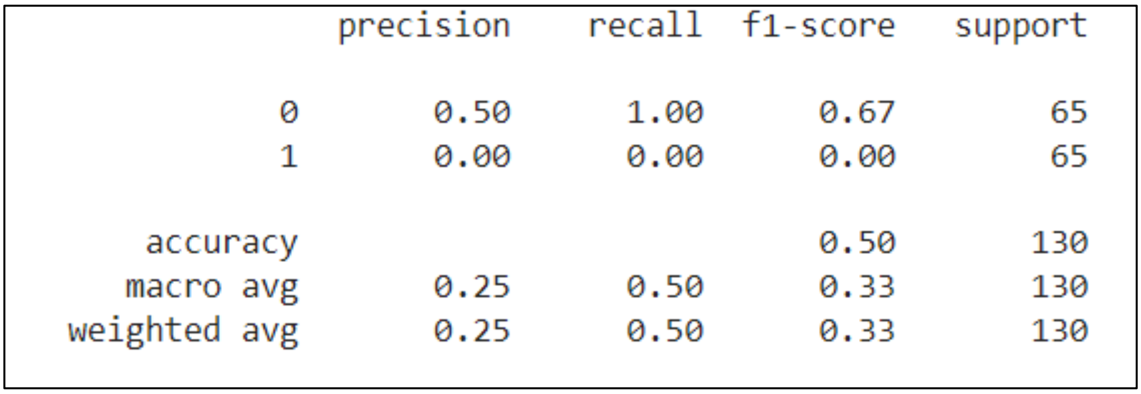


Figure 4.12: Confusion Matrix - Support Vector Classifier with N- gram range (1,2)

4.1.6 Support Vector Classifier with N- gram range (1,3)

Figures 4.13 and 4.14 present the classification report and confusion matrix for the Support Vector Classifier model using count vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        65
           1       0.00      0.00      0.00        65

    accuracy                           0.50       130
   macro avg       0.25      0.50      0.33       130
weighted avg       0.25      0.50      0.33       130
```

Figure 4.13: Classification Report - Support Vector Classifier with N- gram range (1,3)

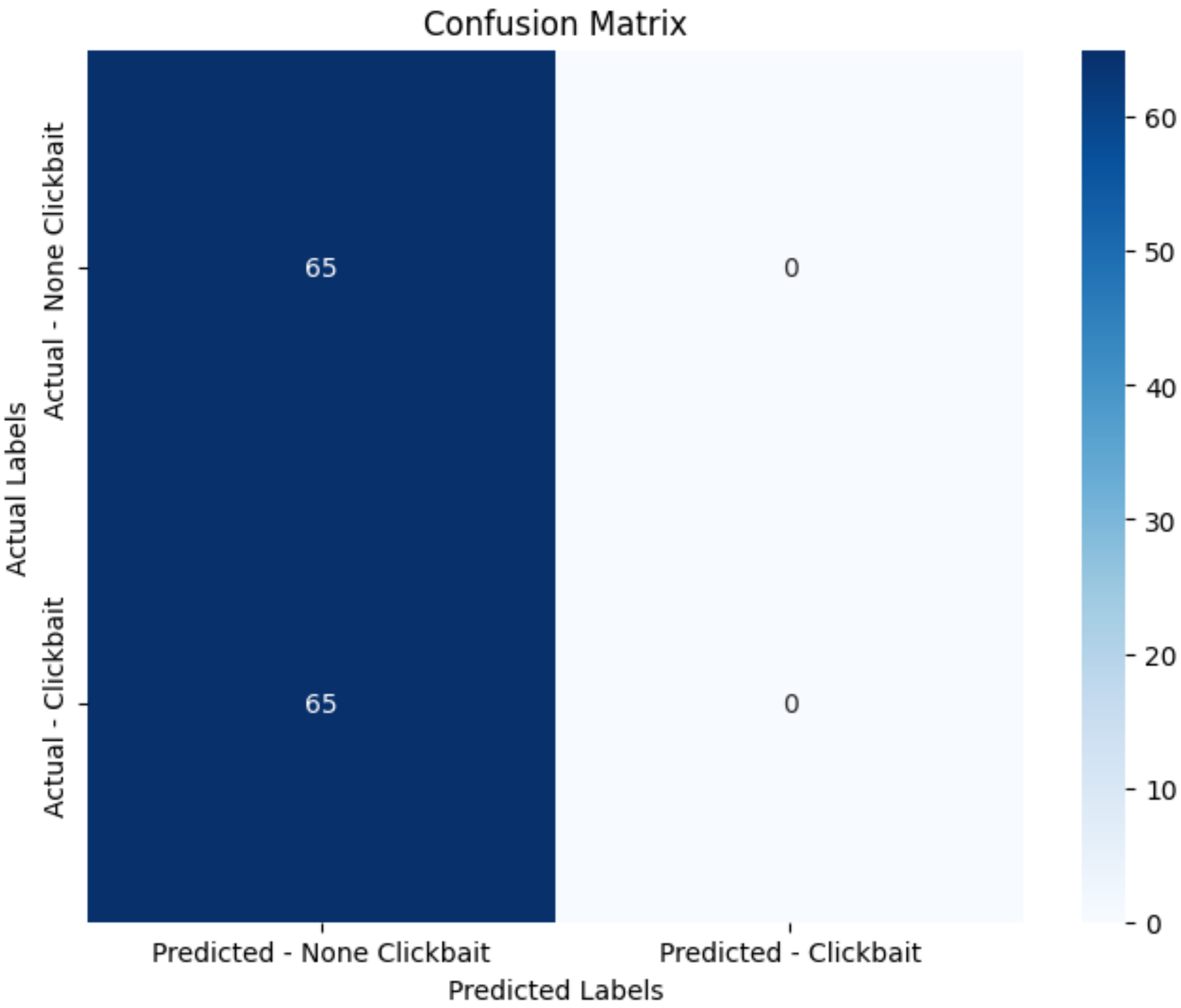
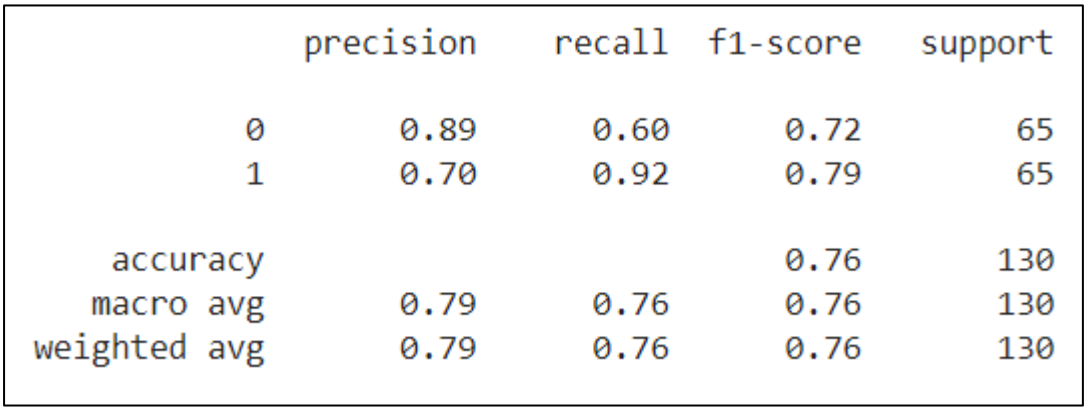


Figure 4.14: Confusion Matrix - Support Vector Classifier with N- gram range (1,3)

4.1.7 Multinomial Naive Bayes Classifier with N- gram range (1,1)

Figures 4.15 and 4.16 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using count vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.89      0.60      0.72        65
           1       0.70      0.92      0.79        65

    accuracy                           0.76       130
   macro avg       0.79      0.76      0.76       130
weighted avg       0.79      0.76      0.76       130
```

Figure 4.15:Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,1)

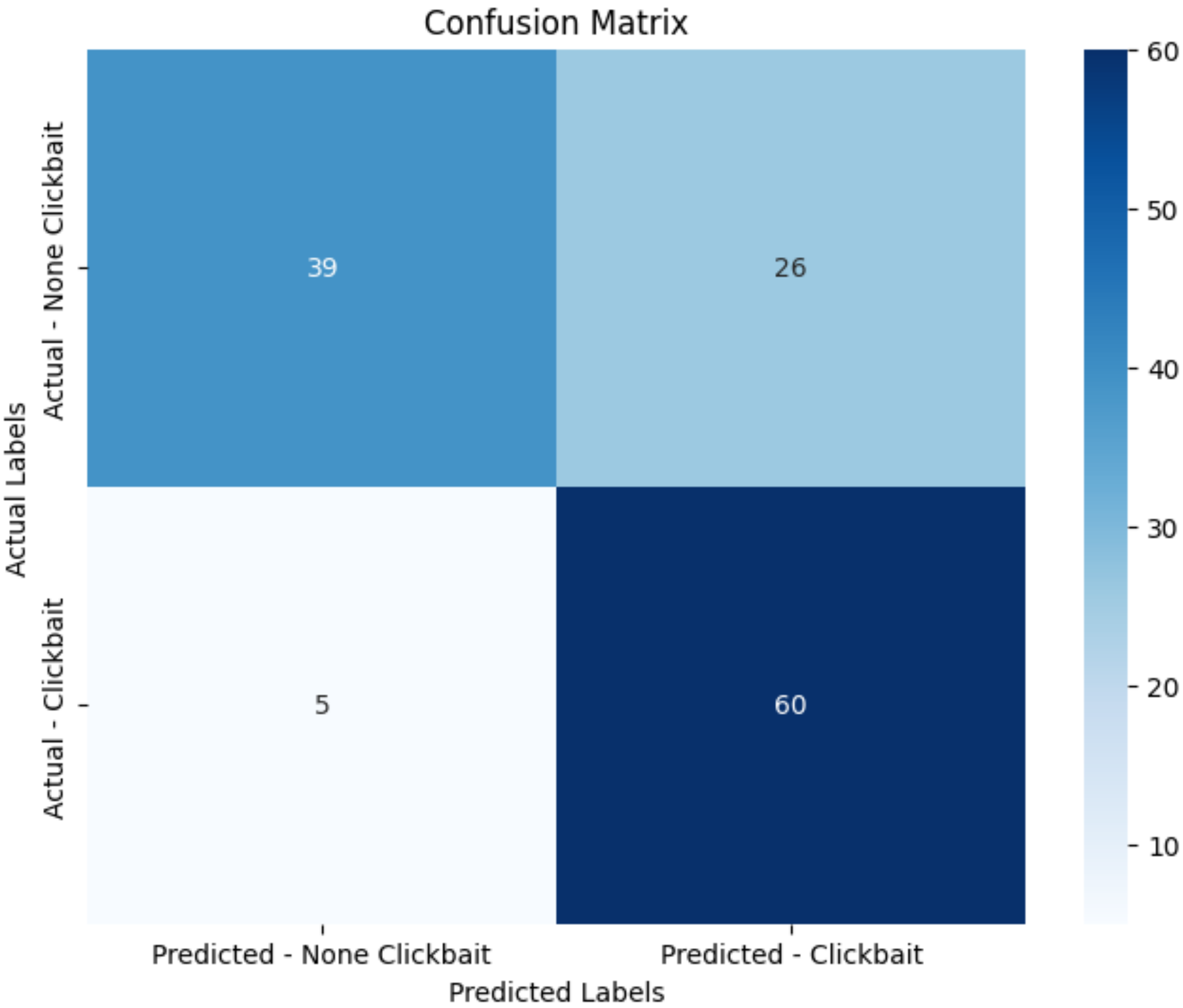
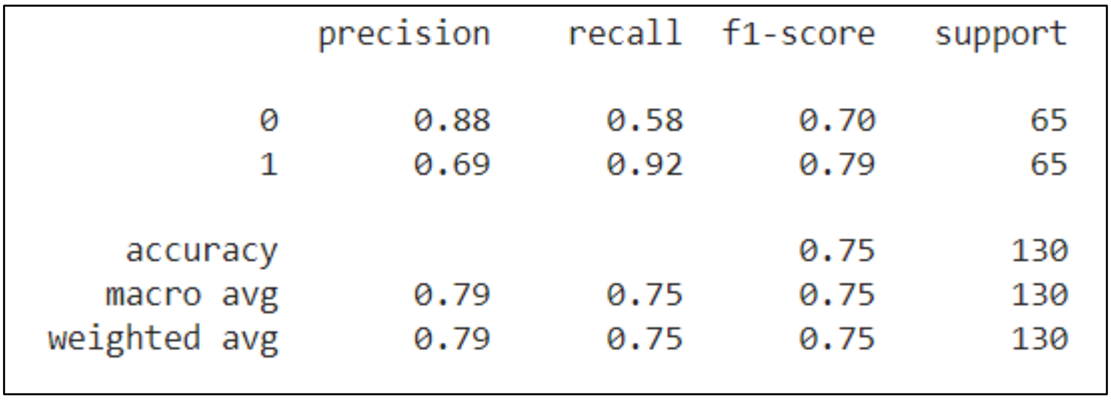


Figure 4.16: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,1)

4.1.8 Multinomial Naive Bayes Classifier with N- gram range (1,2)

Figures 4.17 and 4.18 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using count vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.88      0.58      0.70        65
           1       0.69      0.92      0.79        65

    accuracy                           0.75       130
   macro avg       0.79      0.75      0.75       130
weighted avg       0.79      0.75      0.75       130
```

Figure 4.17: Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,2)

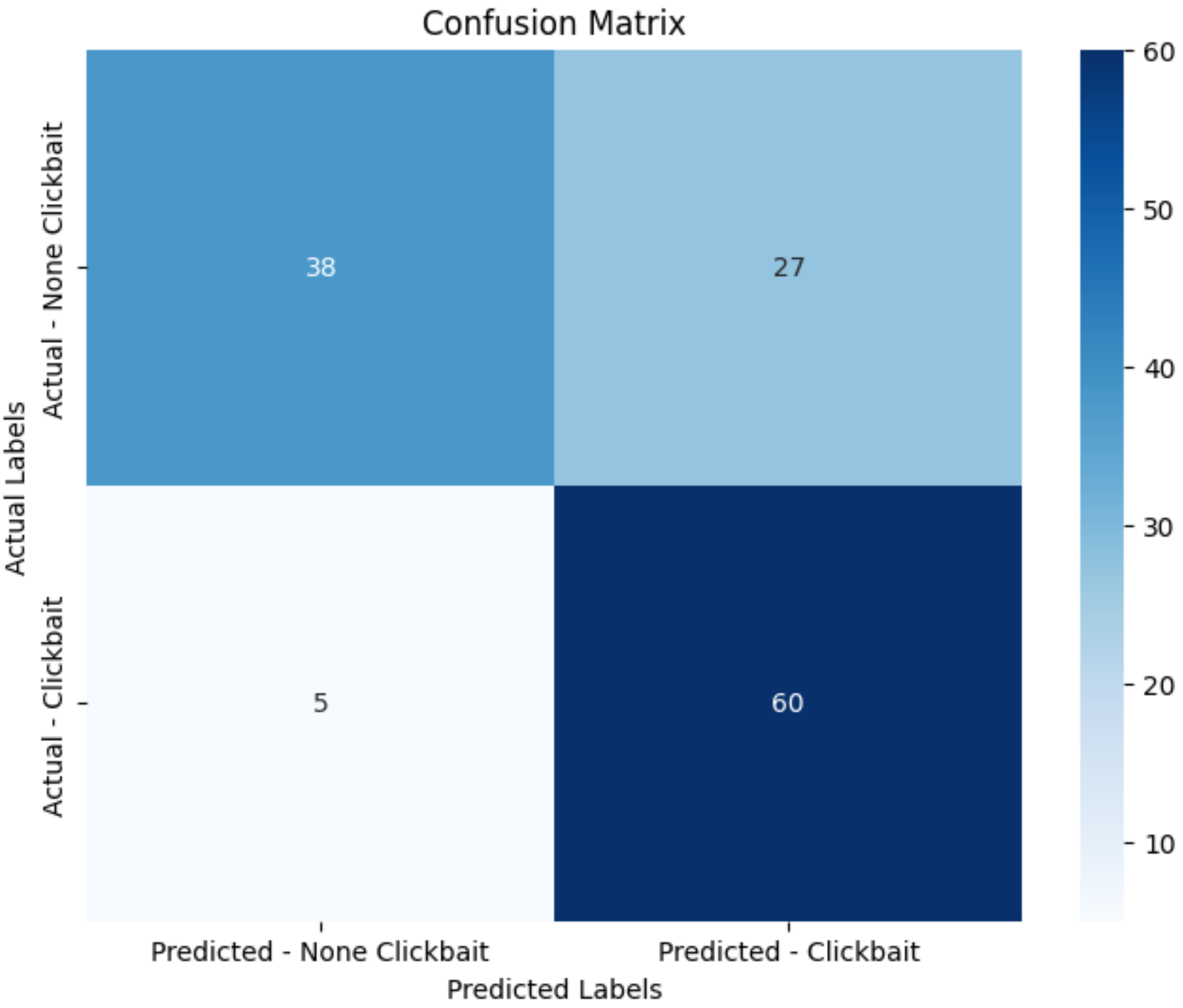
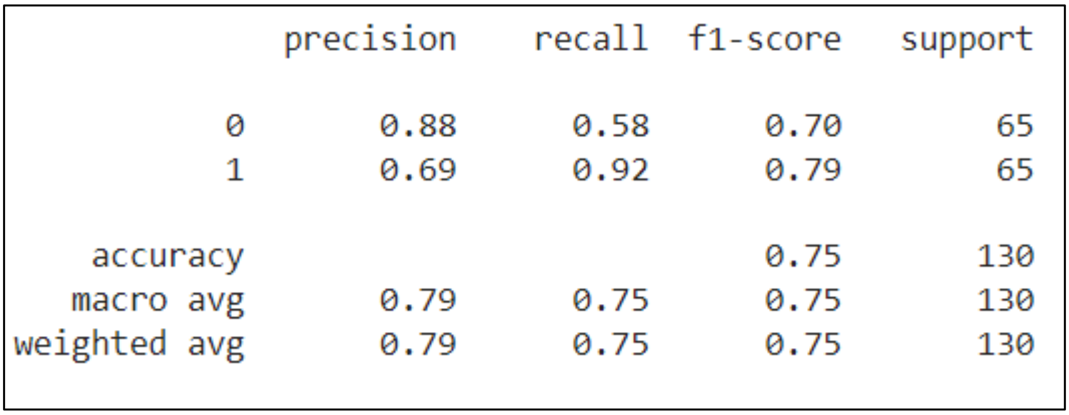


Figure 4.18: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,2)

4.1.9 Multinomial Naive Bayes Classifier with N- gram range (1,3)

Figures 4.19 and 4.20 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using count vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.88      0.58      0.70        65
           1       0.69      0.92      0.79        65

    accuracy                           0.75       130
   macro avg       0.79      0.75      0.75       130
weighted avg       0.79      0.75      0.75       130
```

Figure 4.20: Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,3)

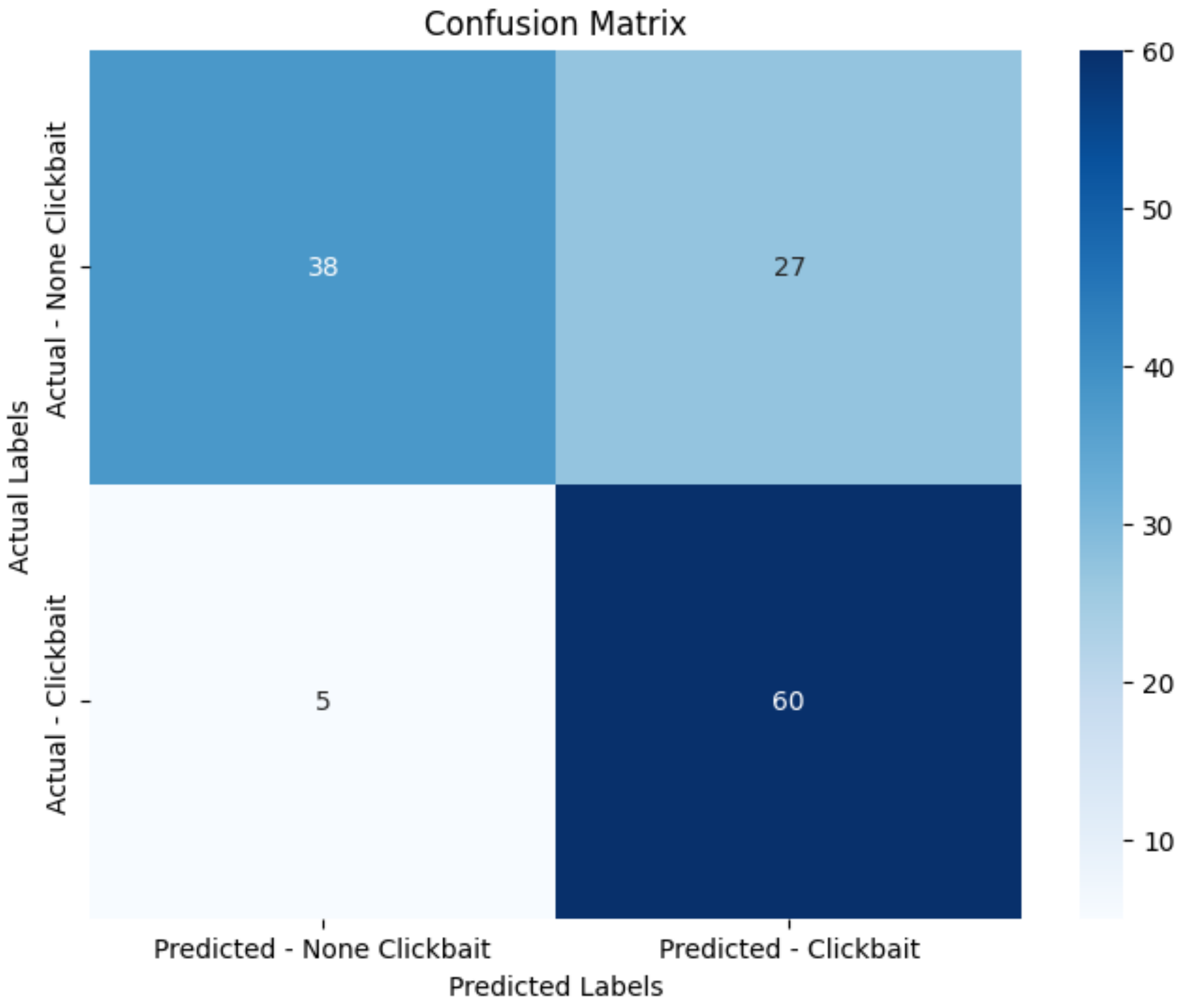
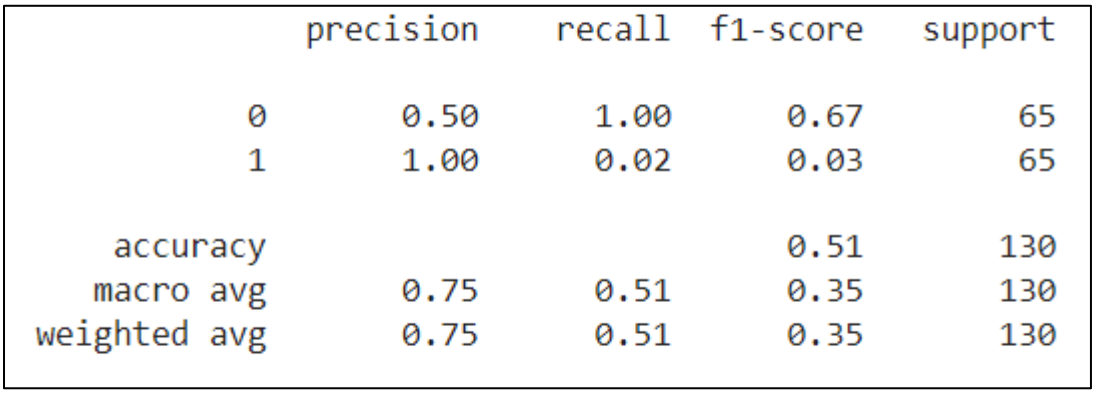


Figure 4.19: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,3)

4.1.10 Kneighbors Classifier with N- gram range (1,1)

Figures 4.21 and 4.22 present the classification report and confusion matrix for the Kneighbors Classifier model using count vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        65
           1       1.00      0.02      0.03        65

    accuracy                           0.51       130
   macro avg       0.75      0.51      0.35       130
weighted avg       0.75      0.51      0.35       130
```

Figure 4.21: Classification Report - Kneighbors Classifier with N- gram range (1,1)

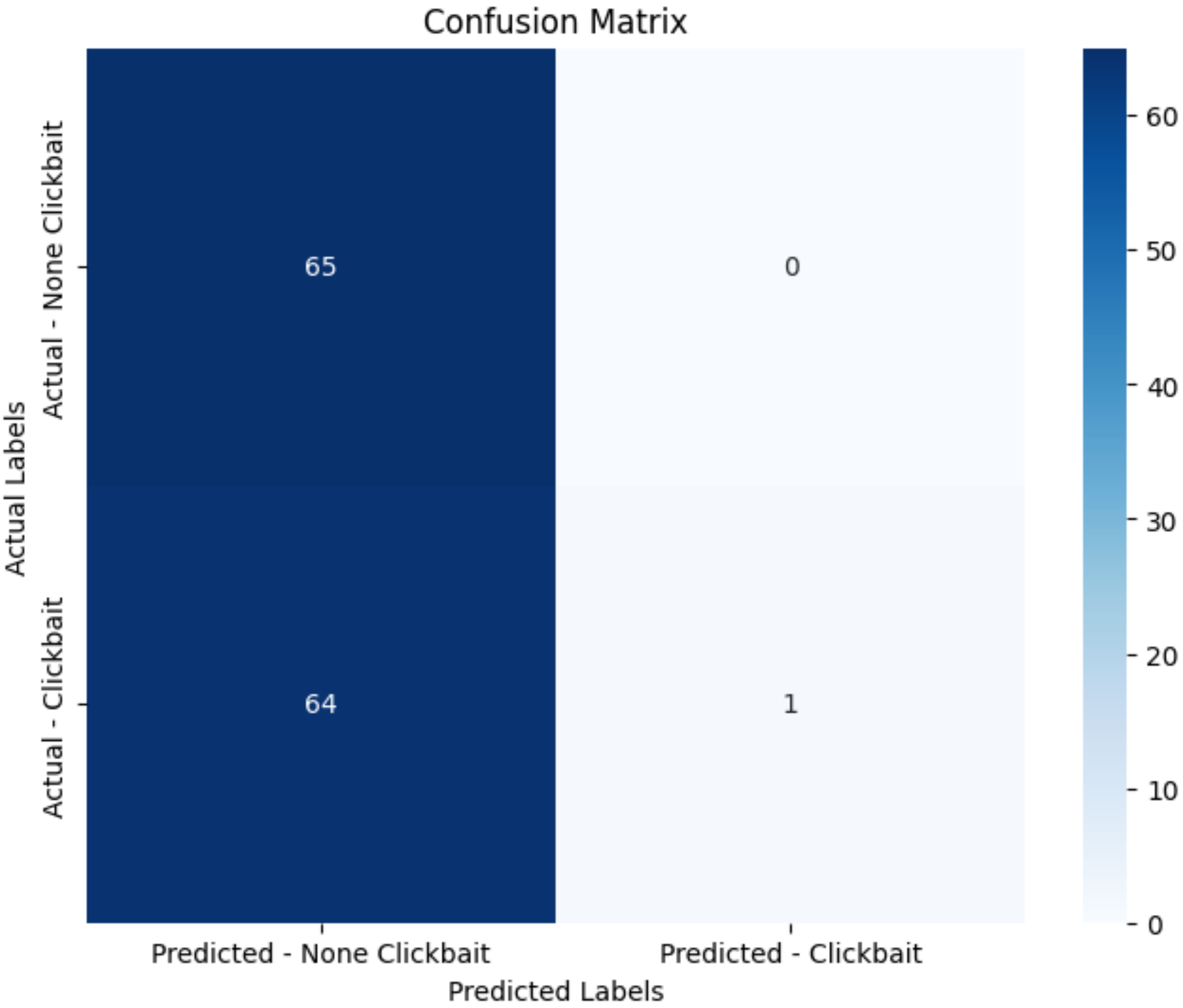
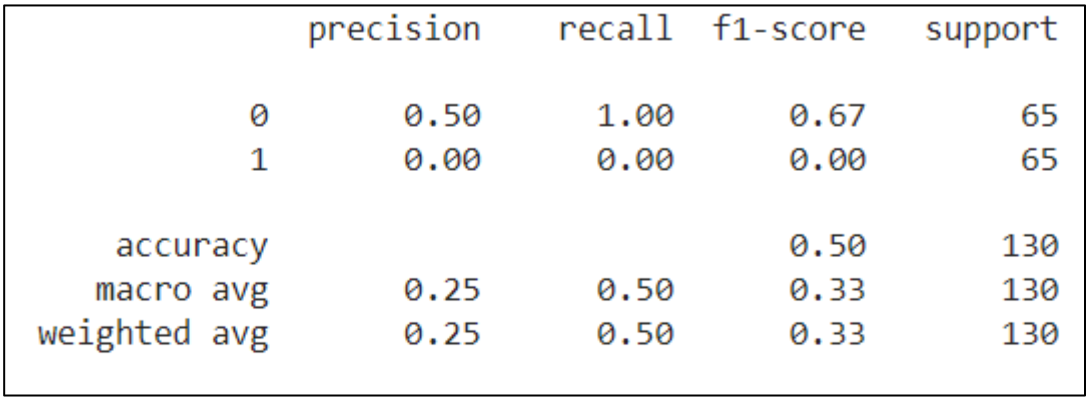


Figure 4.22: Confusion Matrix - Kneighbors Classifier with N- gram range (1,1)

4.1.11 Kneighbors Classifier with N- gram range (1,2)

Figures 4.23 and 4.24 present the classification report and confusion matrix for the Kneighbors Classifier model using count vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        65
           1       0.00      0.00      0.00        65

    accuracy                           0.50       130
   macro avg       0.25      0.50      0.33       130
weighted avg       0.25      0.50      0.33       130
```

Figure 4.23: Classification Report - Kneighbors Classifier with N- gram range (1,2)

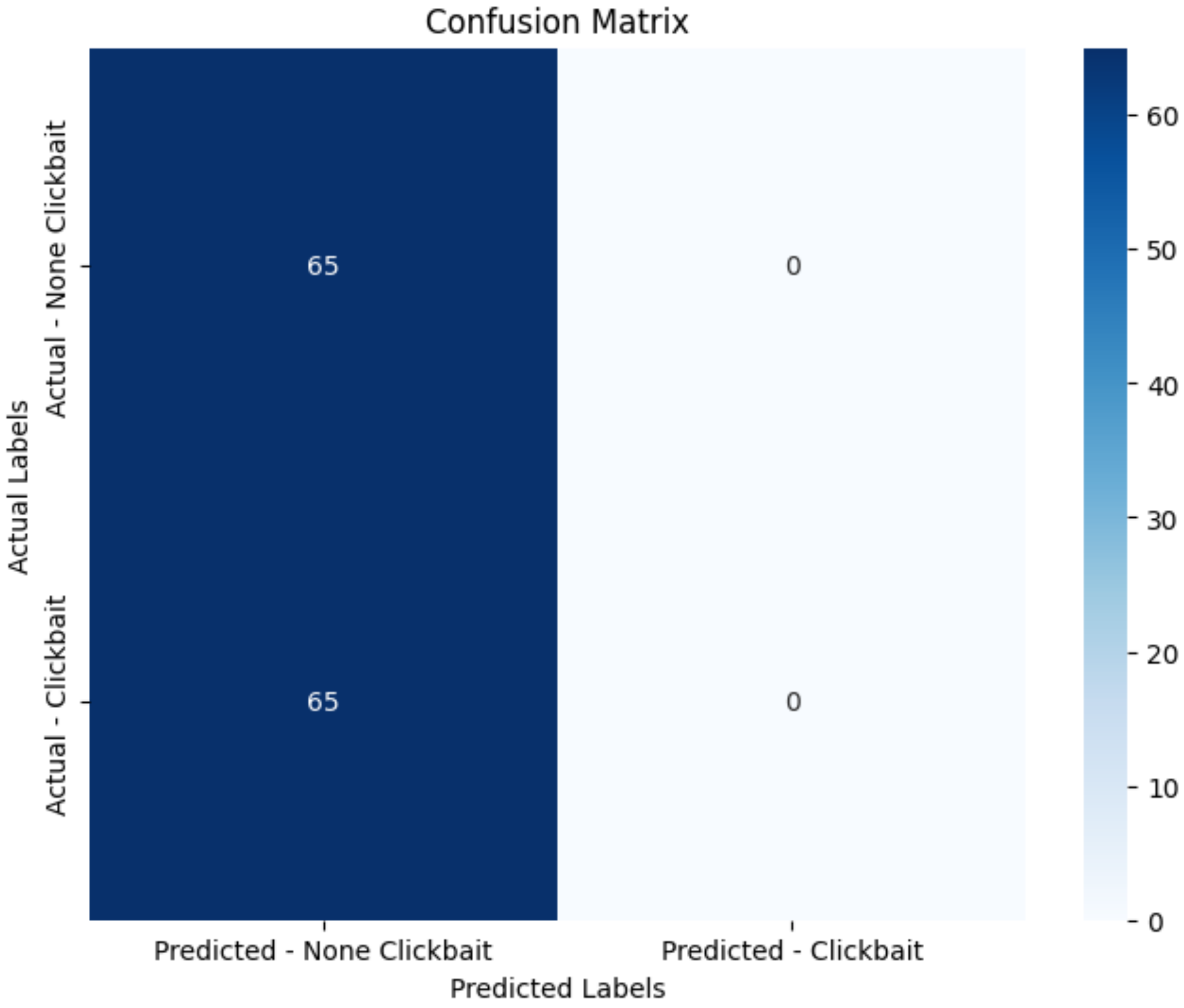
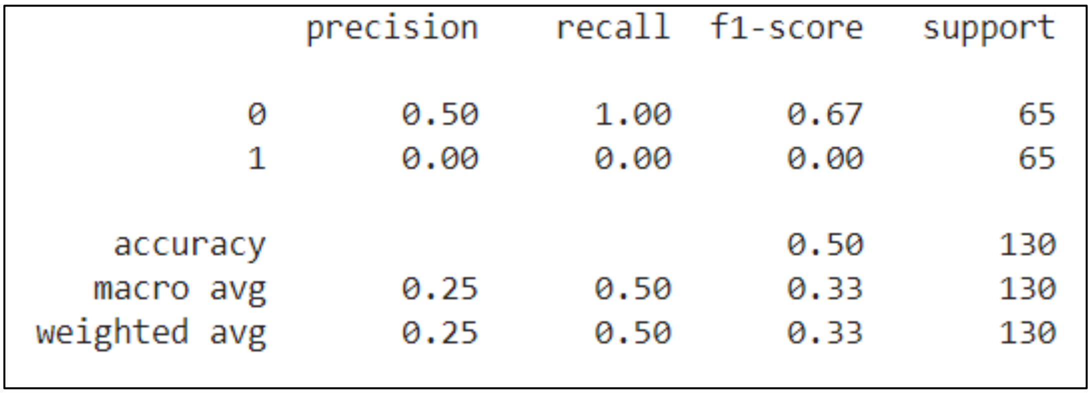


Figure 4.24: Confusion Matrix - Kneighbors Classifier with N- gram range (1,2)

4.1.12 Kneighbors Classifier with N- gram range (1,3)

Figures 4.25 and 4.26 present the classification report and confusion matrix for the Kneighbors Classifier model using count vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67        65
           1       0.00      0.00      0.00        65

    accuracy                           0.50       130
   macro avg       0.25      0.50      0.33       130
weighted avg       0.25      0.50      0.33       130
```

Figure 4.25: Classification Report - Kneighbors Classifier with N- gram range (1,3)

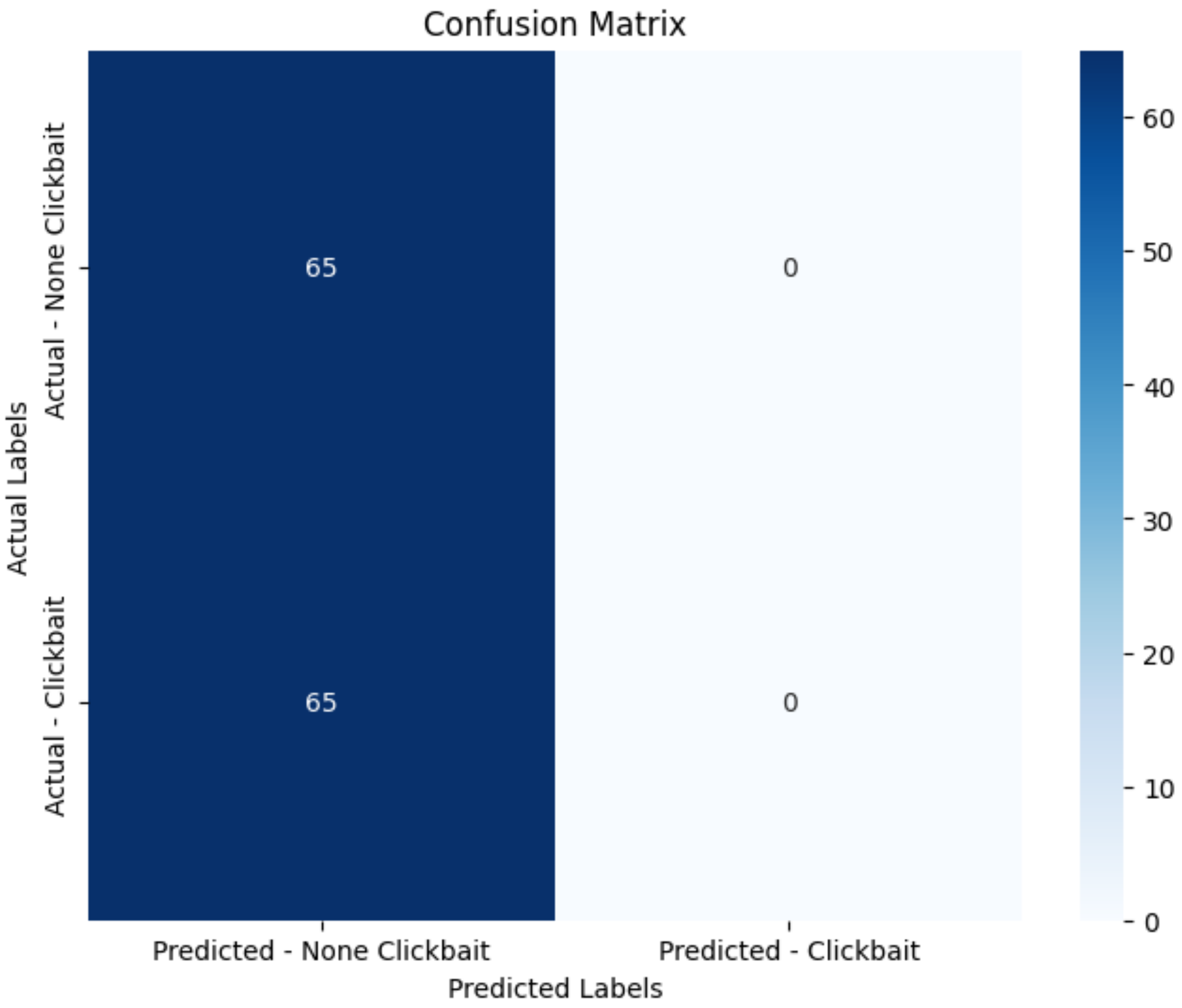
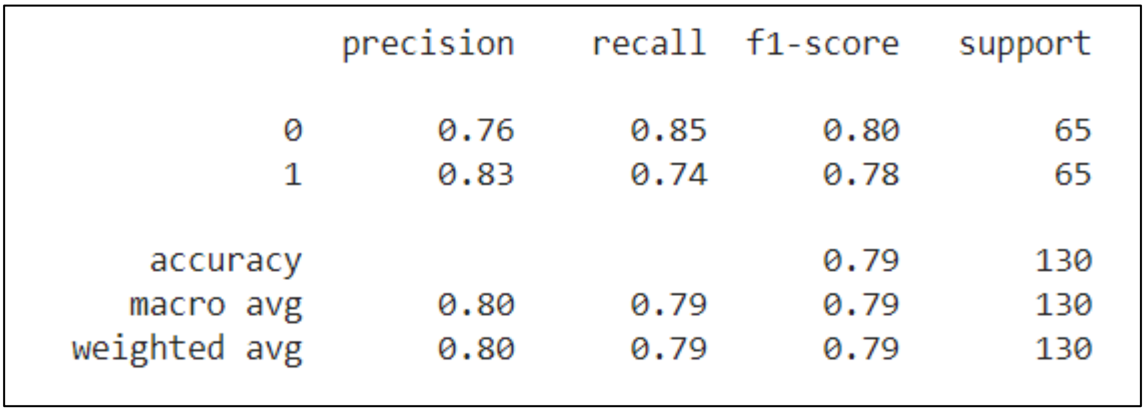


Figure 4.26: Confusion Matrix - Kneighbors Classifier with N- gram range (1,3)

4.2TF-IDF vectorizer

4.2.1 Logistic Regression with N- gram range (1,1)

Figures 4.27 and 4.28 present the classification report and confusion matrix for the Logistic Regression model using TF-IDF vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.76      0.85      0.80        65
           1       0.83      0.74      0.78        65

    accuracy                           0.79       130
   macro avg       0.80      0.79      0.79       130
weighted avg       0.80      0.79      0.79       130
```

Figure 4.27: Classification Report - Logistic Regression with N- gram range (1,1)

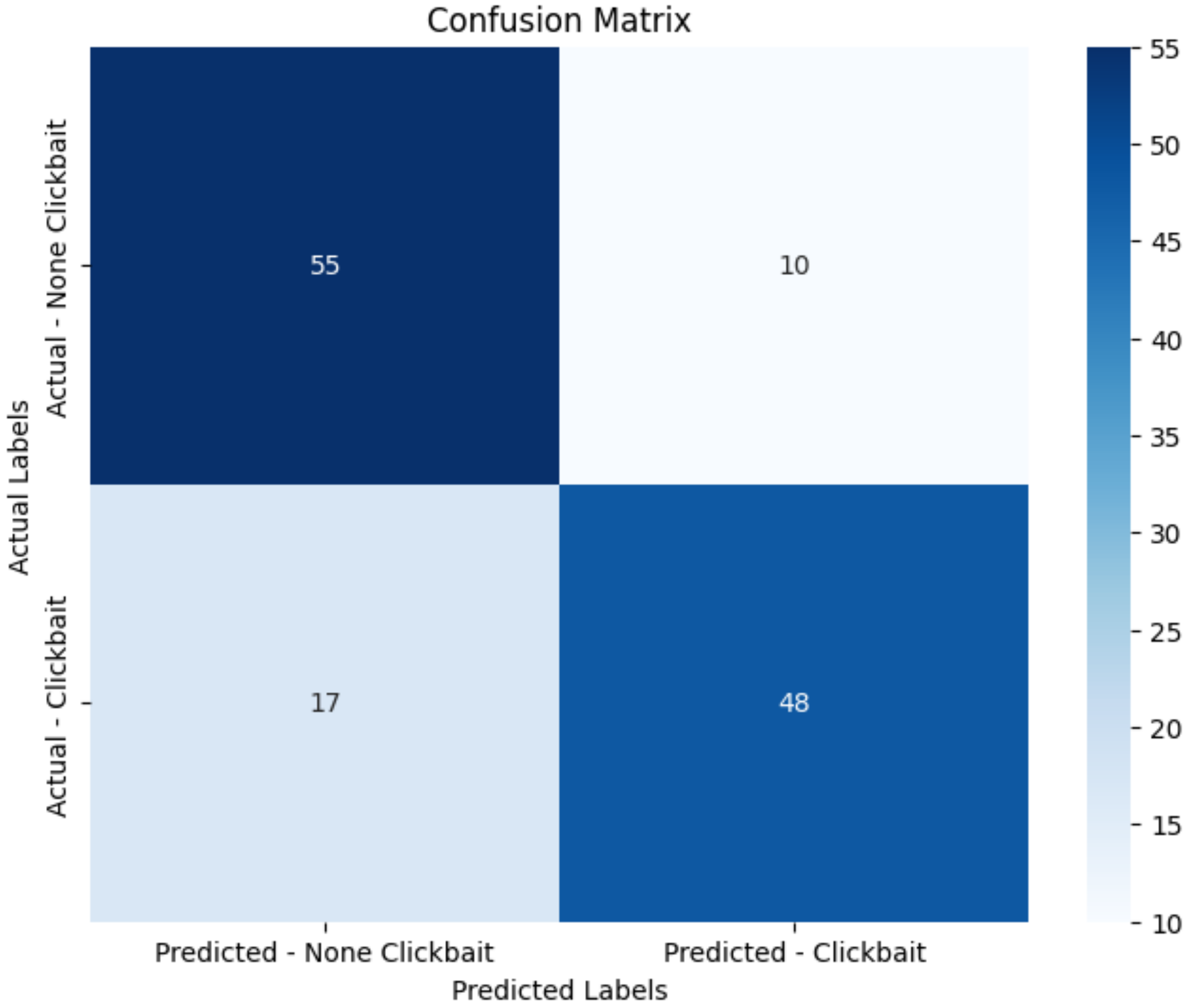
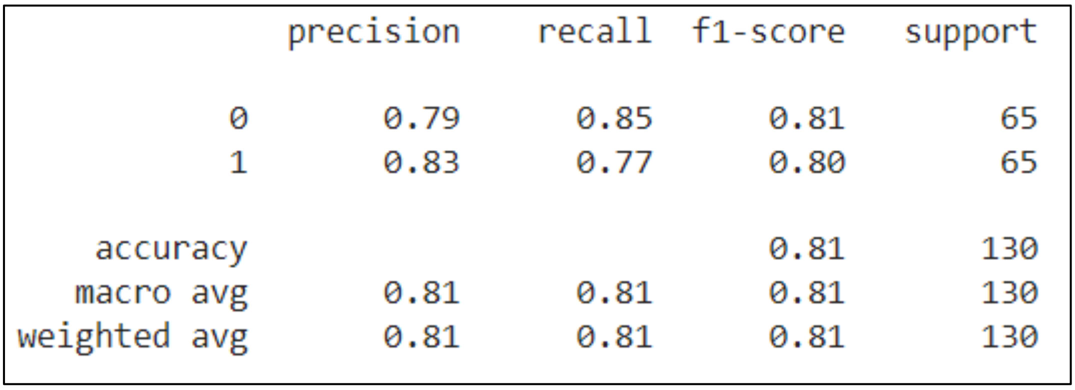


Figure 4.28: Confusion Matrix - Logistic Regression with N- gram range (1,1)

4.2.2 Logistic Regression with N- gram range (1,2)

Figures 4.29 and 4.30 present the classification report and confusion matrix for the Logistic Regression model using TF-IDF vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.79      0.85      0.81        65
           1       0.83      0.77      0.80        65

    accuracy                           0.81       130
   macro avg       0.81      0.81      0.81       130
weighted avg       0.81      0.81      0.81       130
```

Figure 4.30: Classification Report - Logistic Regression with N- gram range (1,2)

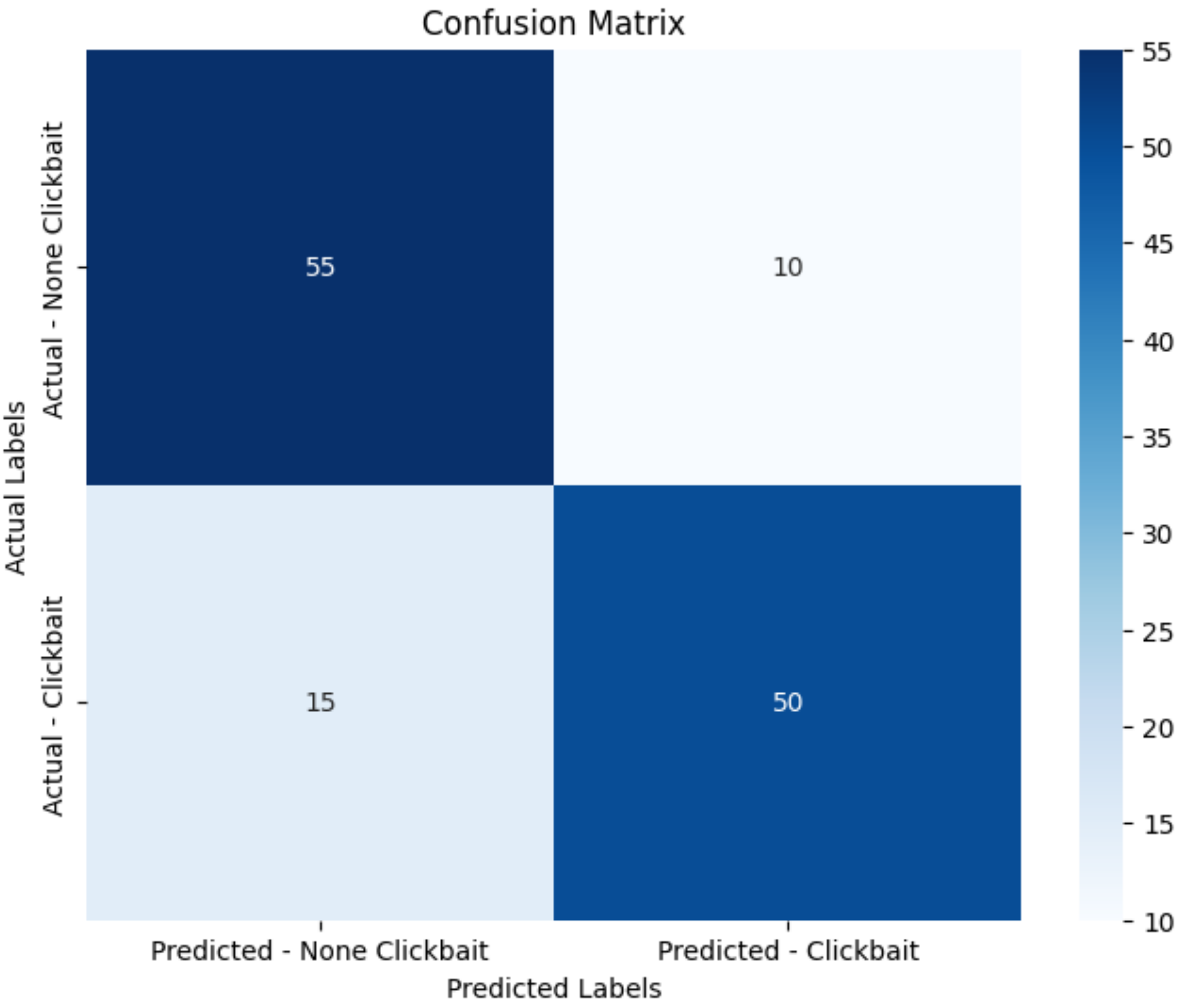
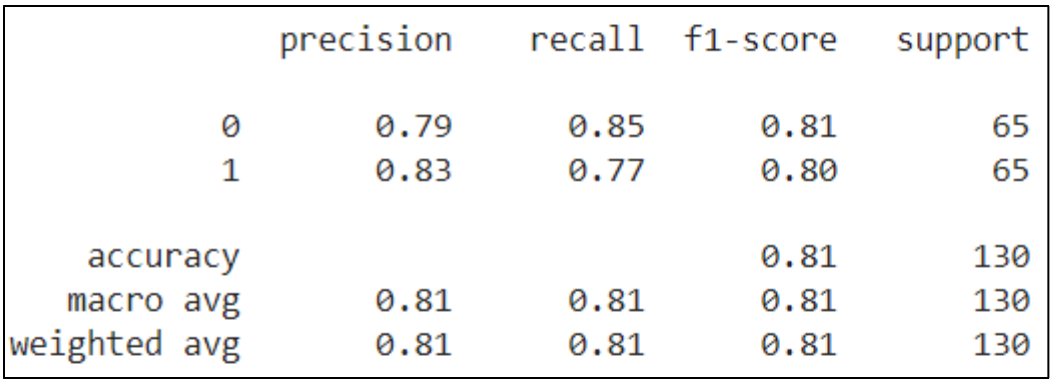


Figure 4.29: Confusion Matrix - Logistic Regression with N- gram range (1,2)

4.2.3 Logistic Regression with N- gram range (1,3)

Figures 4.31 and 4.32 present the classification report and confusion matrix for the Logistic Regression model using TF-IDF vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.79      0.85      0.81        65
           1       0.83      0.77      0.80        65

    accuracy                           0.81       130
   macro avg       0.81      0.81      0.81       130
weighted avg       0.81      0.81      0.81       130
```

Figure 4.31: Classification Report - Logistic Regression with N- gram range (1,3)

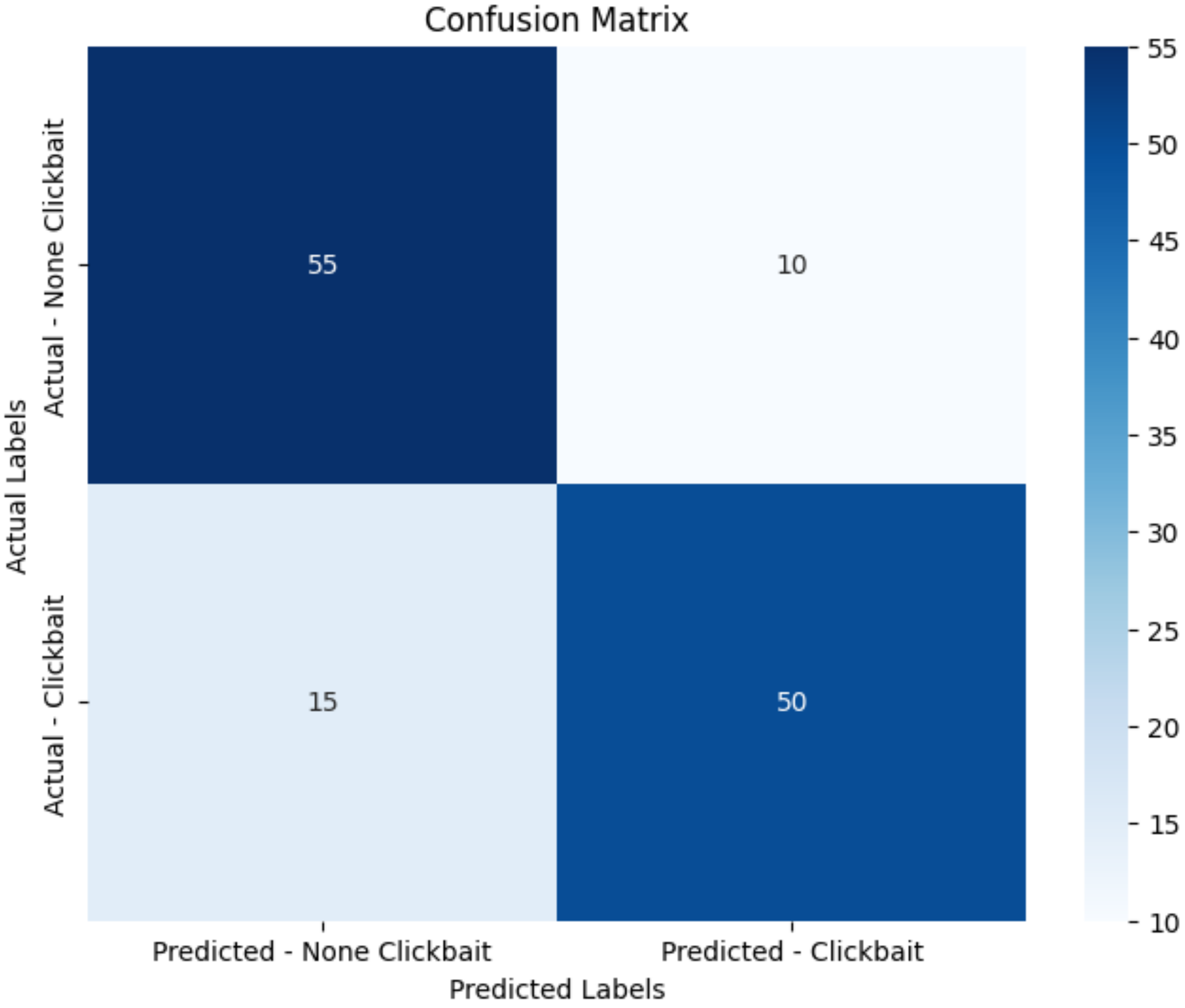
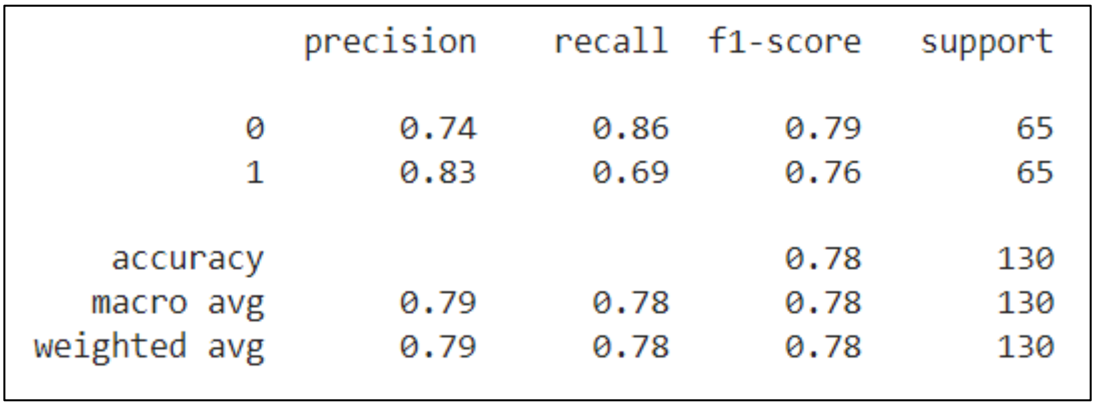


Figure 4.32: Confusion Matrix - Logistic Regression with N- gram range (1,3)

4.2.4 Support Vector Classifier with N- gram range (1,1)

Figures 4.33 and 4.34 present the classification report and confusion matrix for the Support Vector Classifier model using TF-IDF vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.74      0.86      0.79        65
           1       0.83      0.69      0.76        65

    accuracy                           0.78       130
   macro avg       0.79      0.78      0.78       130
weighted avg       0.79      0.78      0.78       130
```

Figure 4.33: Classification Report - Support Vector Classifier with N- gram range (1,1)

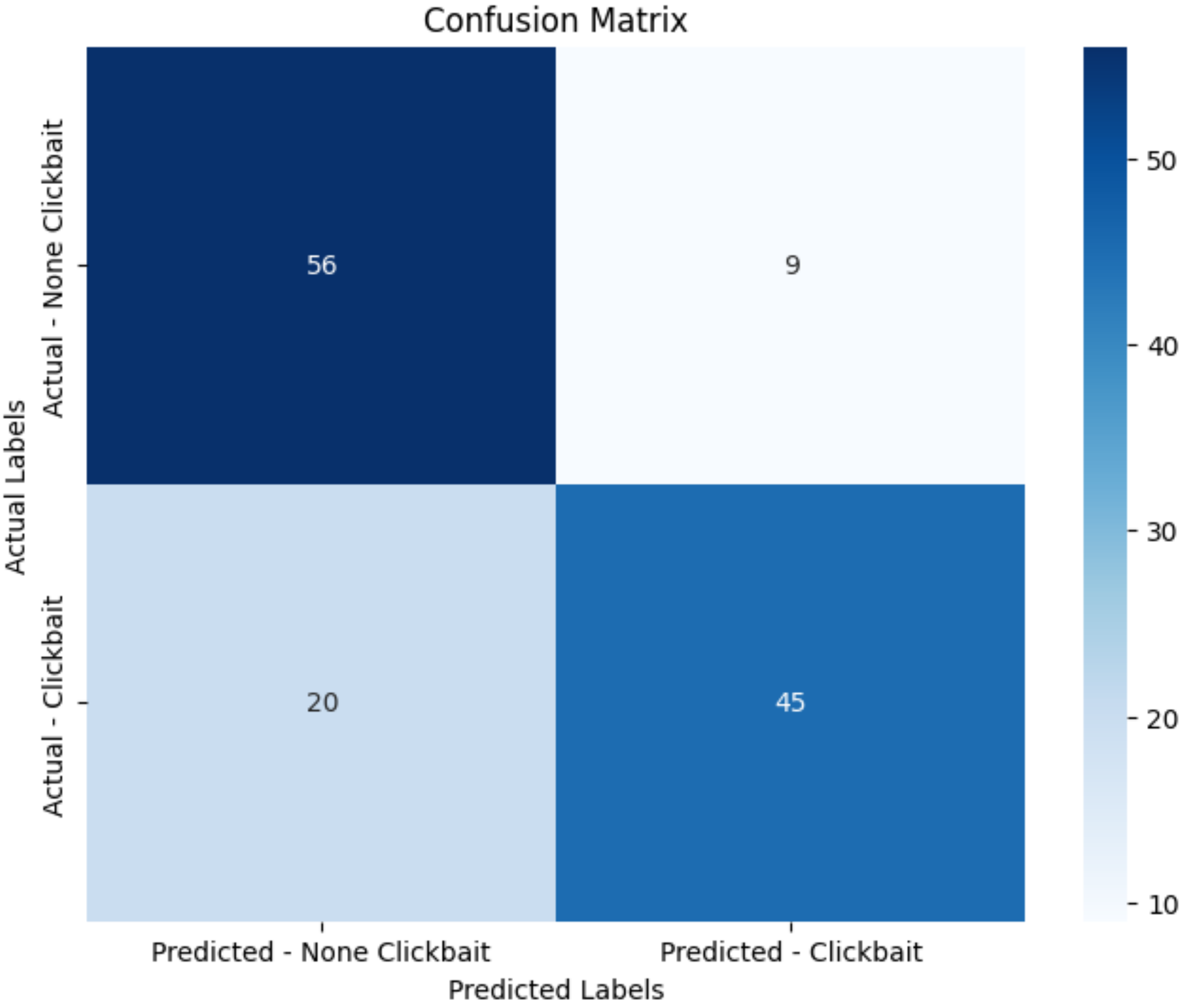
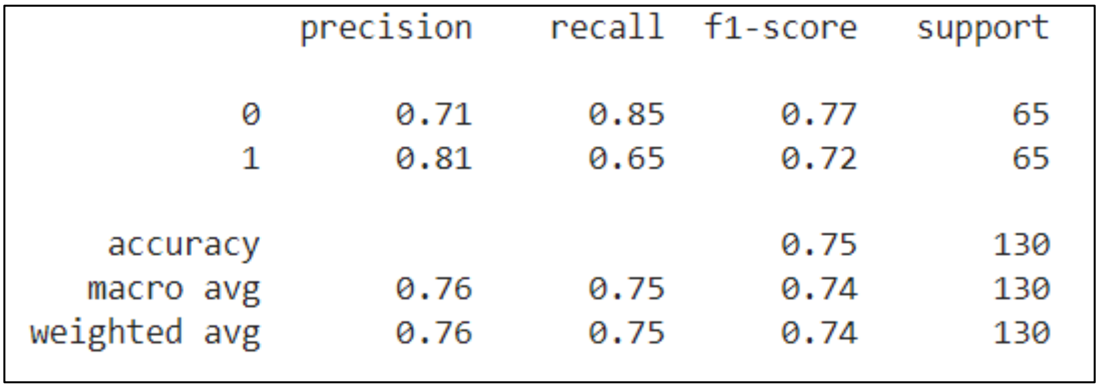


Figure 4.34: Confusion Matrix - Support Vector Classifier with N- gram range (1,1)

4.2.5 Support Vector Classifier with N- gram range (1,2)

Figures 4.35 and 4.36 present the classification report and confusion matrix for the Support Vector Classifier model using TF-IDF vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.71      0.85      0.77        65
           1       0.81      0.65      0.72        65

    accuracy                           0.75       130
   macro avg       0.76      0.75      0.74       130
weighted avg       0.76      0.75      0.74       130
```

Figure 4.35: Classification Report - Support Vector Classifier with N- gram range (1,2)

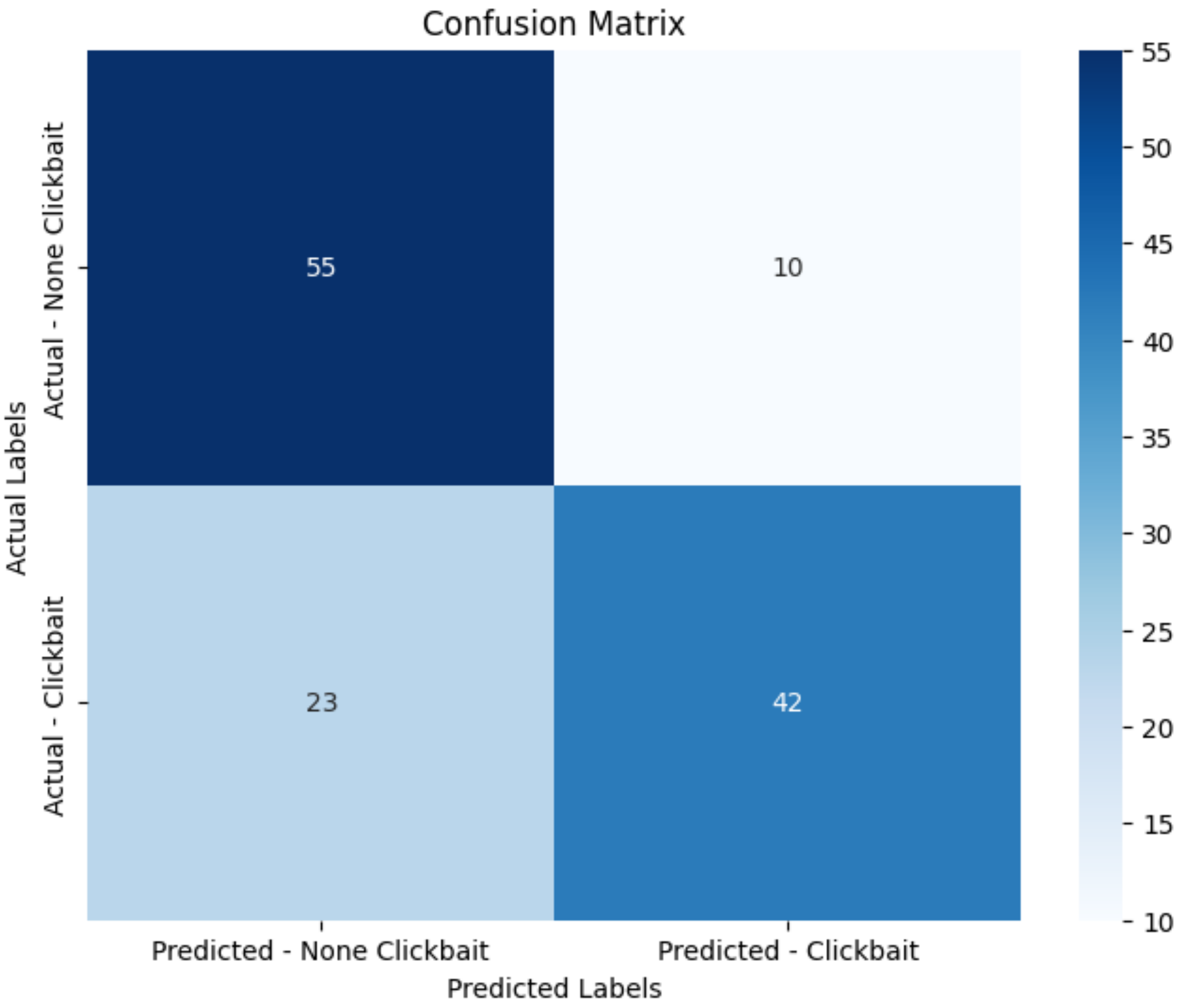
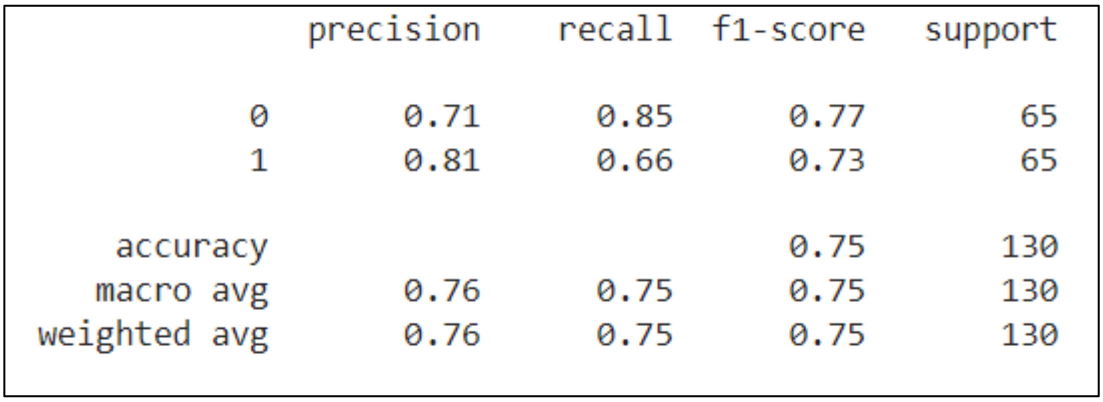


Figure 4.36: Confusion Matrix - Support Vector Classifier with N- gram range (1,2)

4.2.6 Support Vector Classifier with N- gram range (1,3)

Figures 4.37 and 4.38 present the classification report and confusion matrix for the Support Vector Classifier model using TF-IDF vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.71      0.85      0.77        65
           1       0.81      0.66      0.73        65

    accuracy                           0.75       130
   macro avg       0.76      0.75      0.75       130
weighted avg       0.76      0.75      0.75       130
```

Figure 4.37: Classification Report - Support Vector Classifier with N- gram range (1,3)

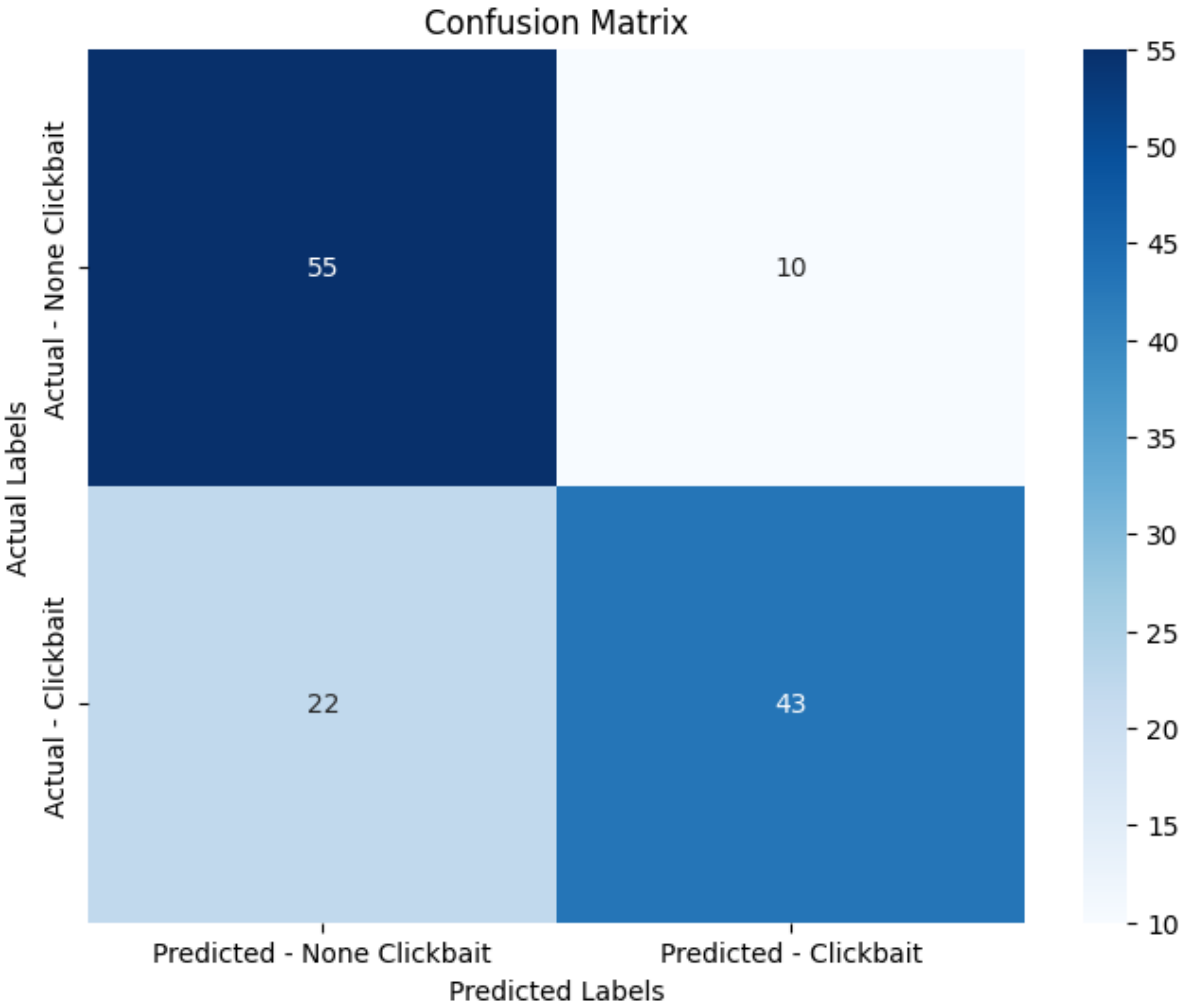
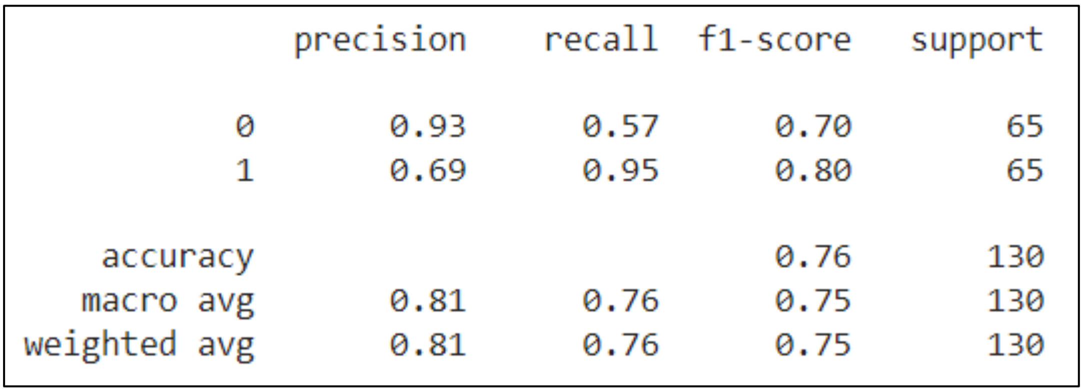


Figure 4.38: Confusion Matrix - Support Vector Classifier with N- gram range (1,3)

4.2.7 Multinomial Naive Bayes Classifier with N- gram range (1,1)

Figures 4.39 and 4.40 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using TF-IDF vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.93      0.57      0.70        65
           1       0.69      0.95      0.80        65

    accuracy                           0.76       130
   macro avg       0.81      0.76      0.75       130
weighted avg       0.81      0.76      0.75       130
```

Figure 4.39: Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,1)

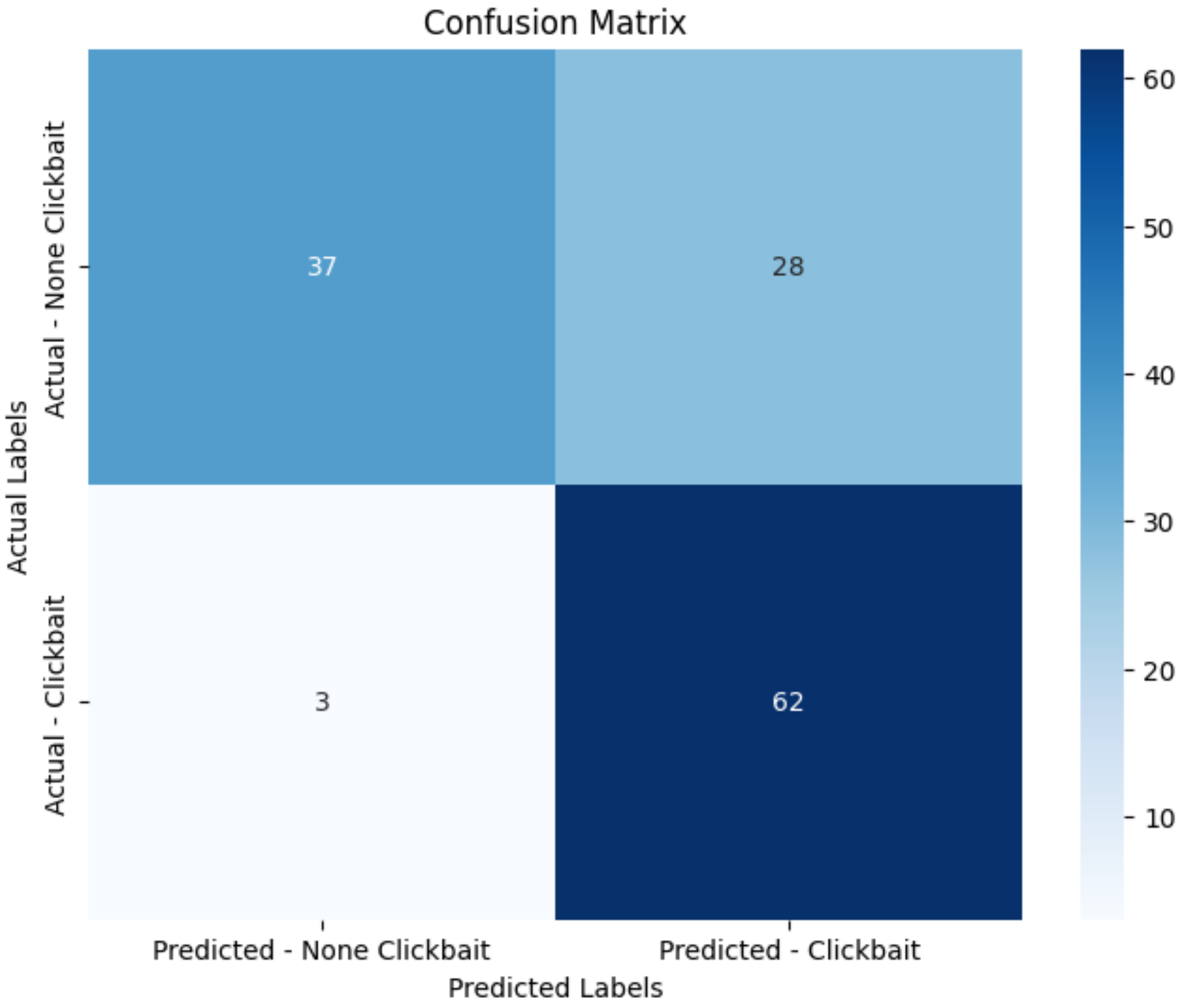
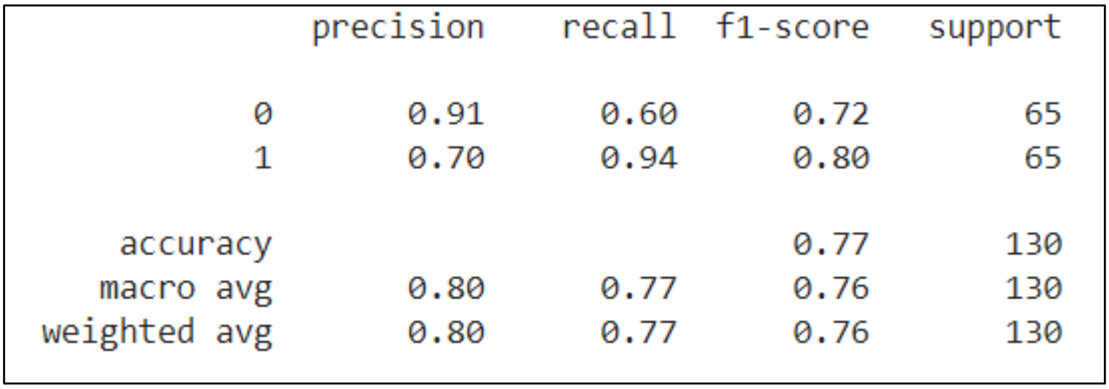


Figure 4.40: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,1)

4.2.8 Multinomial Naive Bayes Classifier with N- gram range (1,2)

Figures 4.41 and 4.42 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using TF-IDF vectorizer technique with N-gram range (1,2).



```
              precision    recall  f1-score   support

           0       0.91      0.60      0.72        65
           1       0.70      0.94      0.80        65

    accuracy                           0.77       130
   macro avg       0.80      0.77      0.76       130
weighted avg       0.80      0.77      0.76       130
```

Figure 4.41: Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,2)

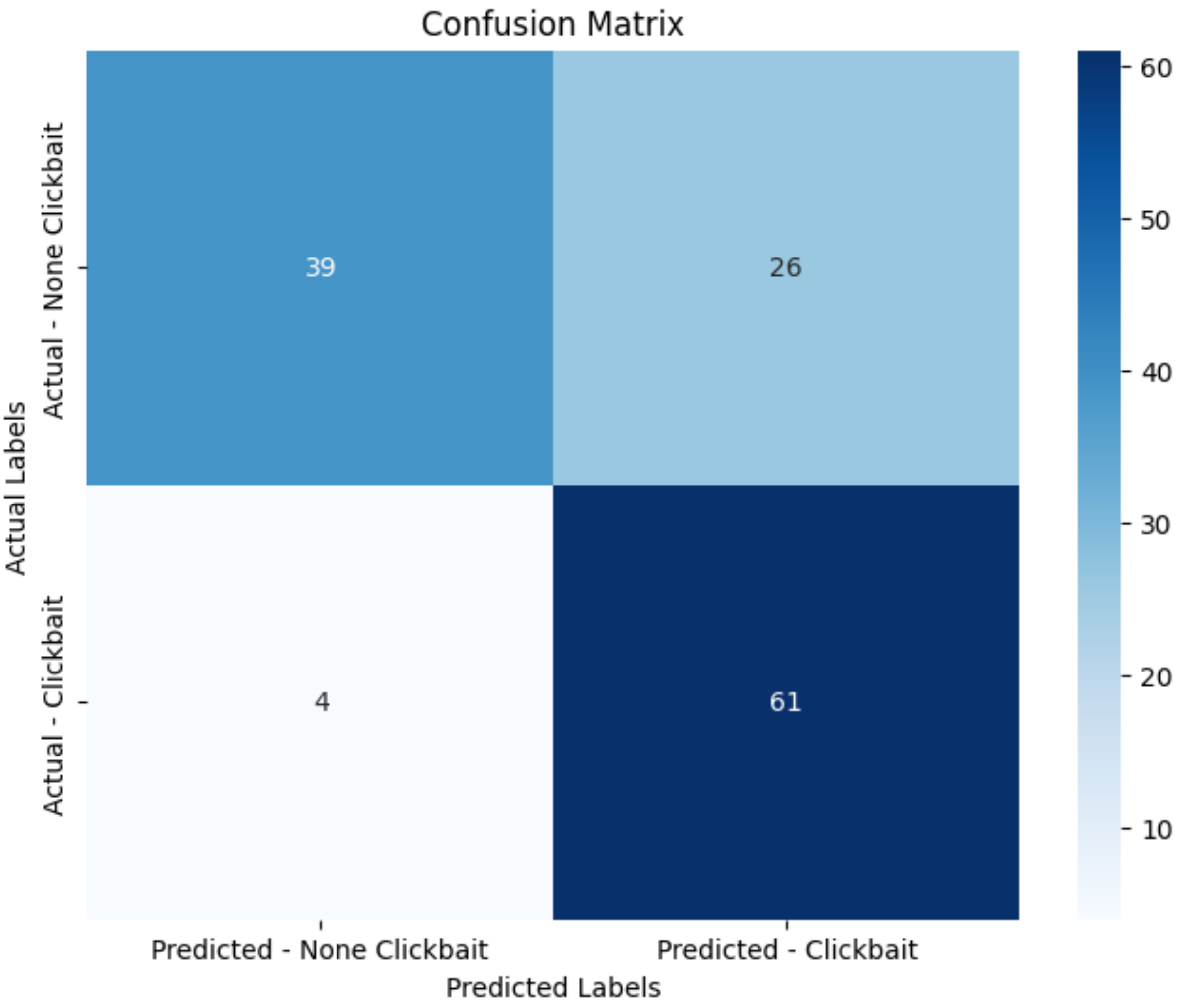
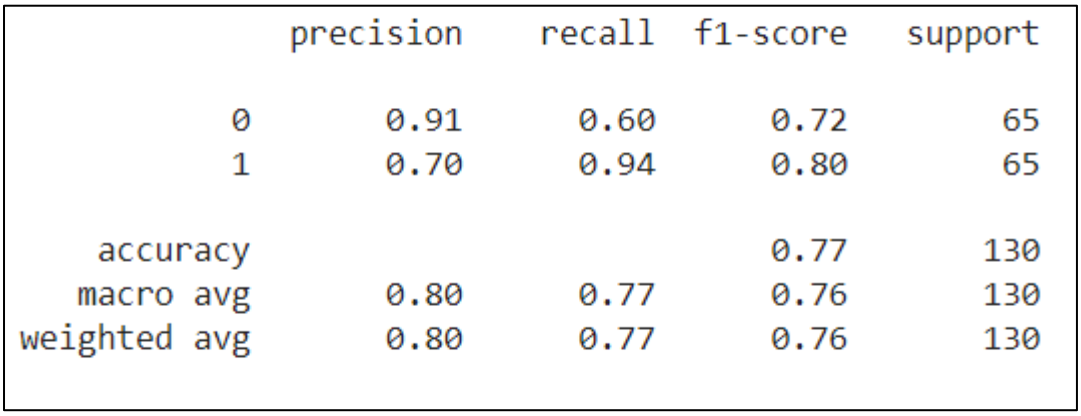


Figure 4.42: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,2)

4.2.9 Multinomial Naive Bayes Classifier with N- gram range (1,3)

Figures 4.43 and 4.44 present the classification report and confusion matrix for the Multinomial Naive Bayes Classifier model using TF-IDF vectorizer technique with N-gram range (1,3).



```
              precision    recall  f1-score   support

           0       0.91      0.60      0.72        65
           1       0.70      0.94      0.80        65

    accuracy                           0.77       130
   macro avg       0.80      0.77      0.76       130
weighted avg       0.80      0.77      0.76       130
```

Figure 4.44: Classification Report - Multinomial Naive Bayes Classifier with N- gram range (1,3)

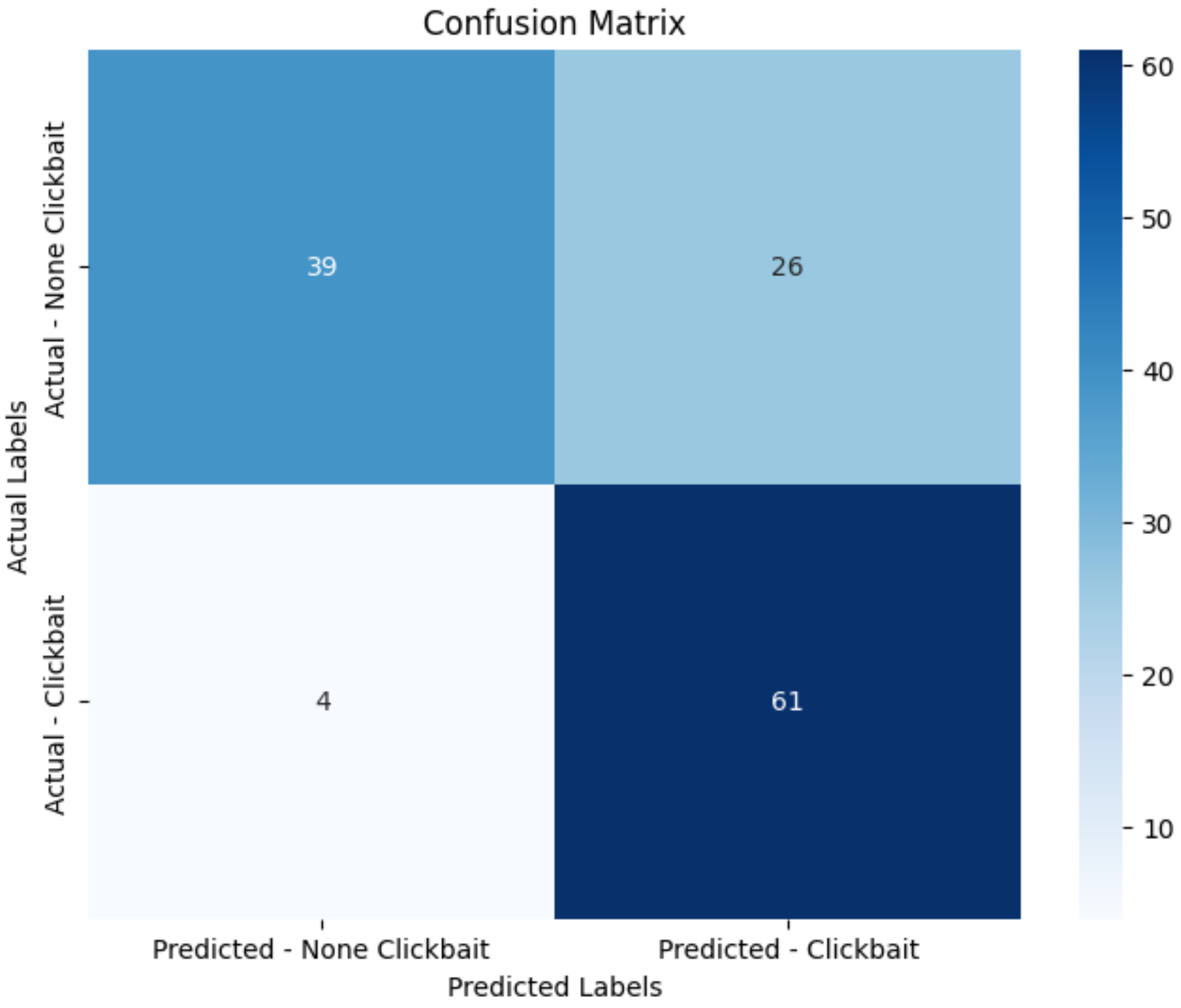
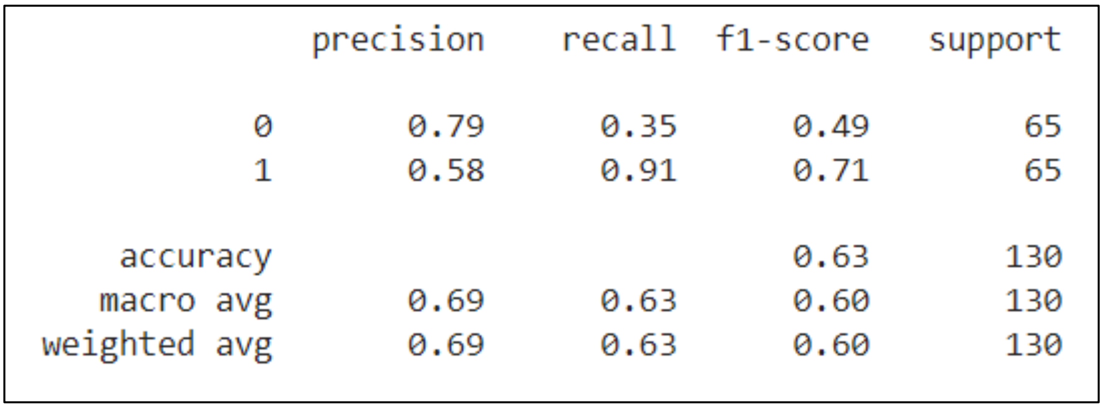


Figure 4.43: Confusion Matrix - Multinomial Naive Bayes Classifier with N- gram range (1,3)

4.2.10 Kneighbors Classifier with N- gram range (1,1)

Figures 4.45 and 4.46 present the classification report and confusion matrix for the Kneighbors Classifier model using TF-IDF vectorizer technique with N-gram range (1,1).



```
              precision    recall  f1-score   support

           0       0.79      0.35      0.49        65
           1       0.58      0.91      0.71        65

    accuracy                           0.63       130
   macro avg       0.69      0.63      0.60       130
weighted avg       0.69      0.63      0.60       130
```

Figure 4.45: Classification Report - Kneighbors Classifier with N- gram range (1,1)

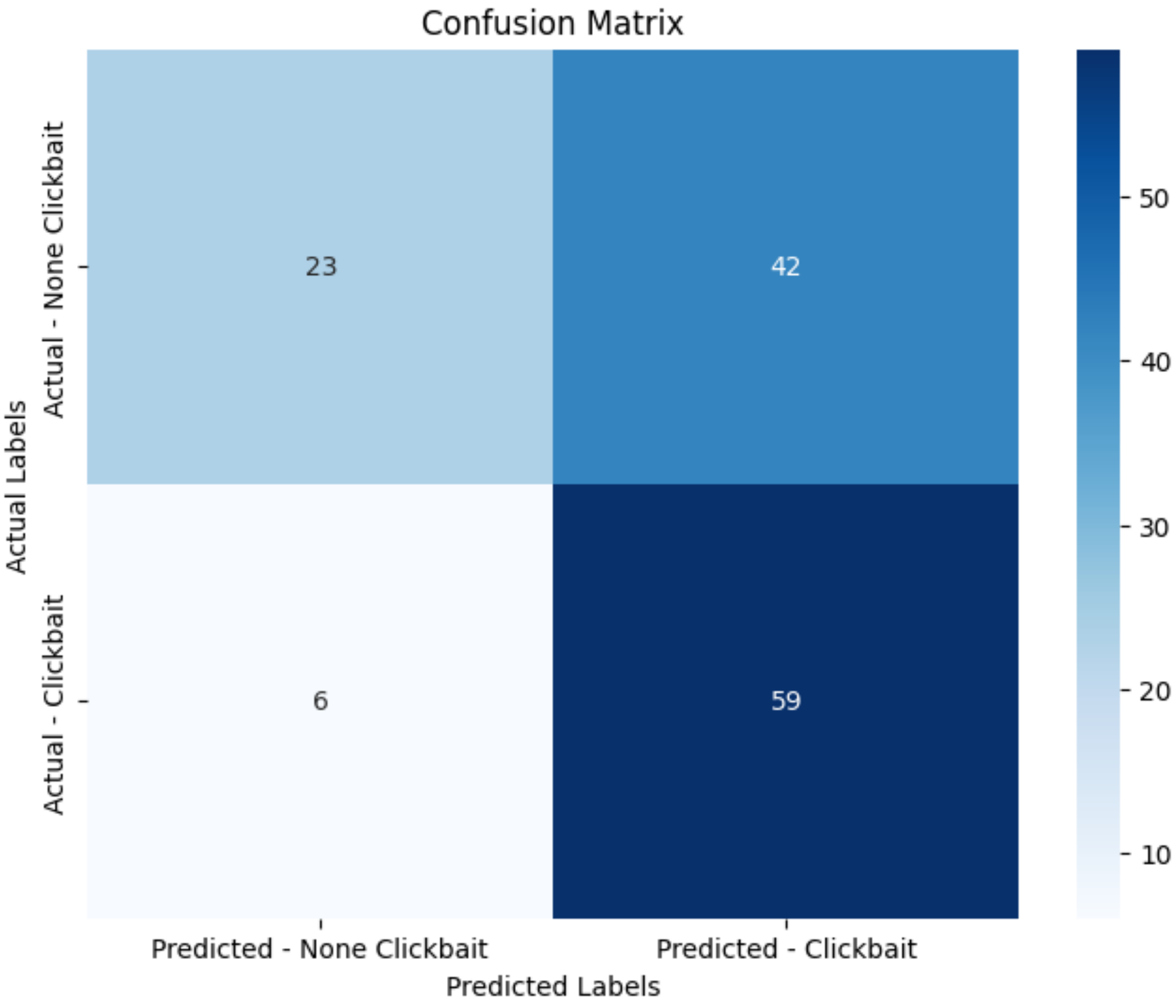
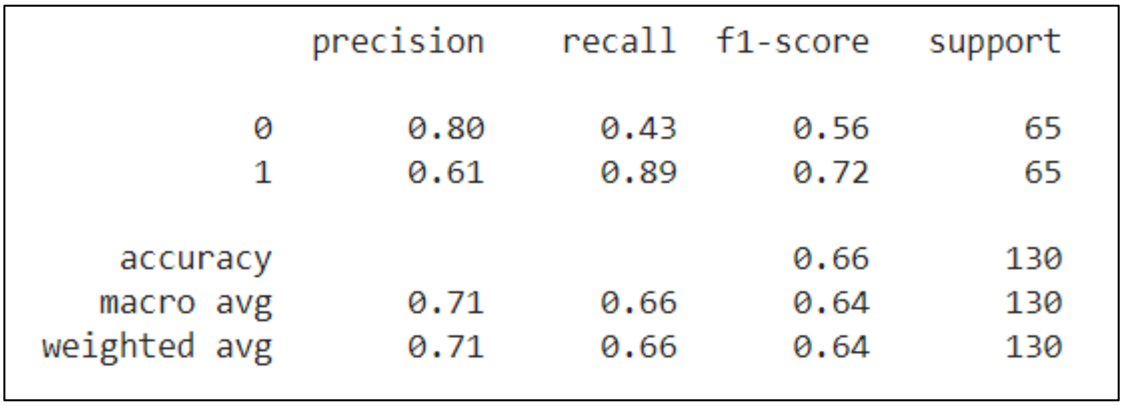


Figure 4.46: Confusion Matrix - Kneighbors Classifier with N- gram range (1,1)

4.2.11 Kneighbors Classifier with N- gram range (1,2)

Figures 4.47 and 4.48 present the classification report and confusion matrix for the Kneighbors Classifier model using TF-IDF vectorizer technique with N-gram range (1,2).

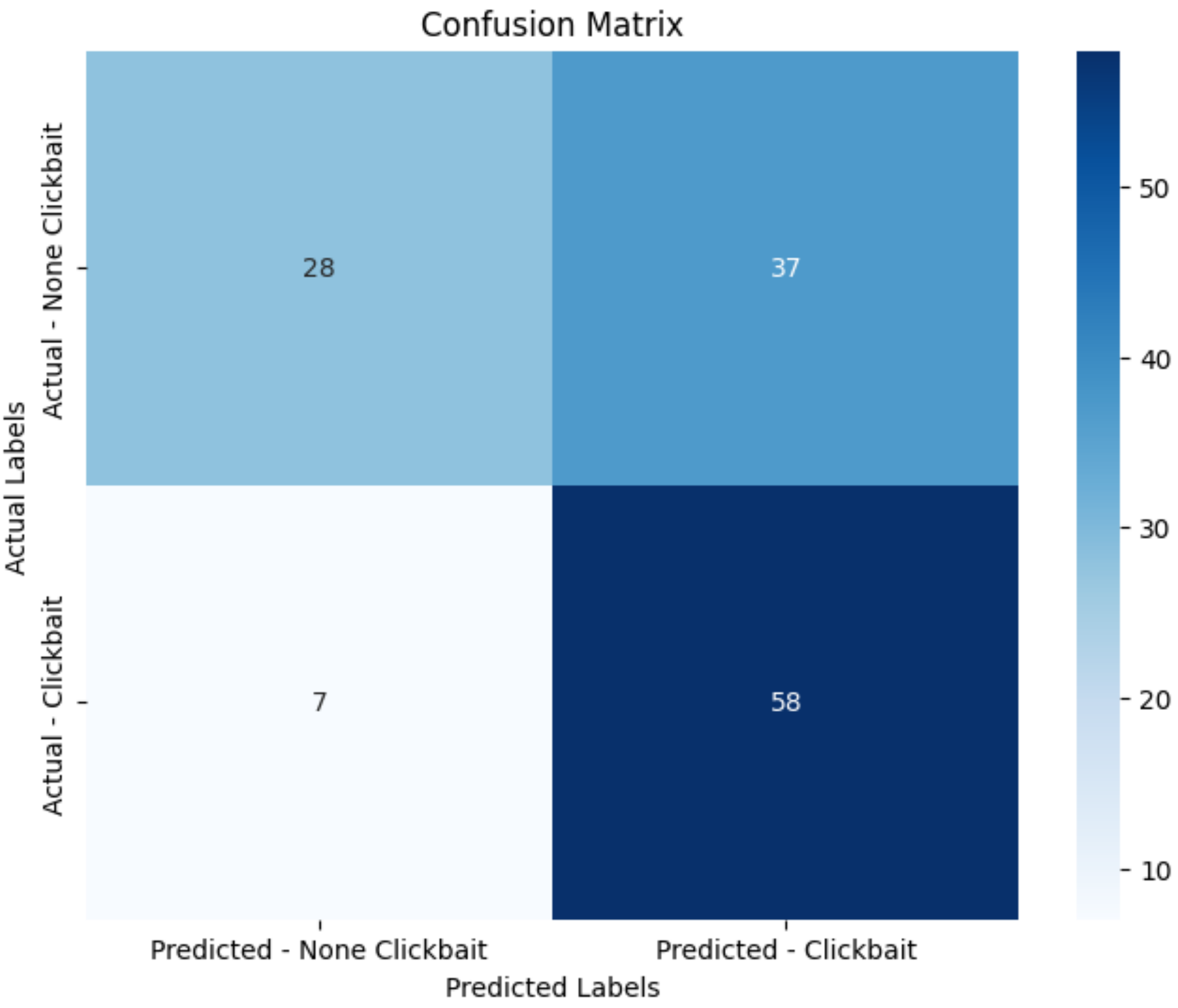


```
              precision    recall  f1-score   support

           0       0.80      0.43      0.56        65
           1       0.61      0.89      0.72        65

    accuracy                           0.66       130
   macro avg       0.71      0.66      0.64       130
weighted avg       0.71      0.66      0.64       130
```

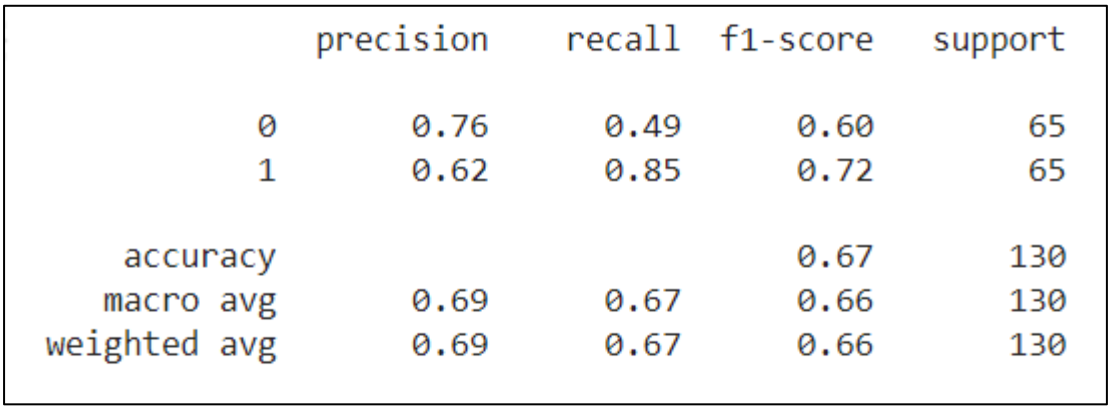
Figure 4.47: Classification Report - Kneighbors Classifier with N- gram range (1,2)Figure



4.48: Confusion Matrix - Kneighbors Classifier with N- gram range (1,2)

4.2.12 Kneighbors Classifier with N- gram range (1,3)

Figures 4.49 and 4.50 present the classification report and confusion matrix for the Kneighbors Classifier model using TF-IDF vectorizer technique with N-gram range (1,3).

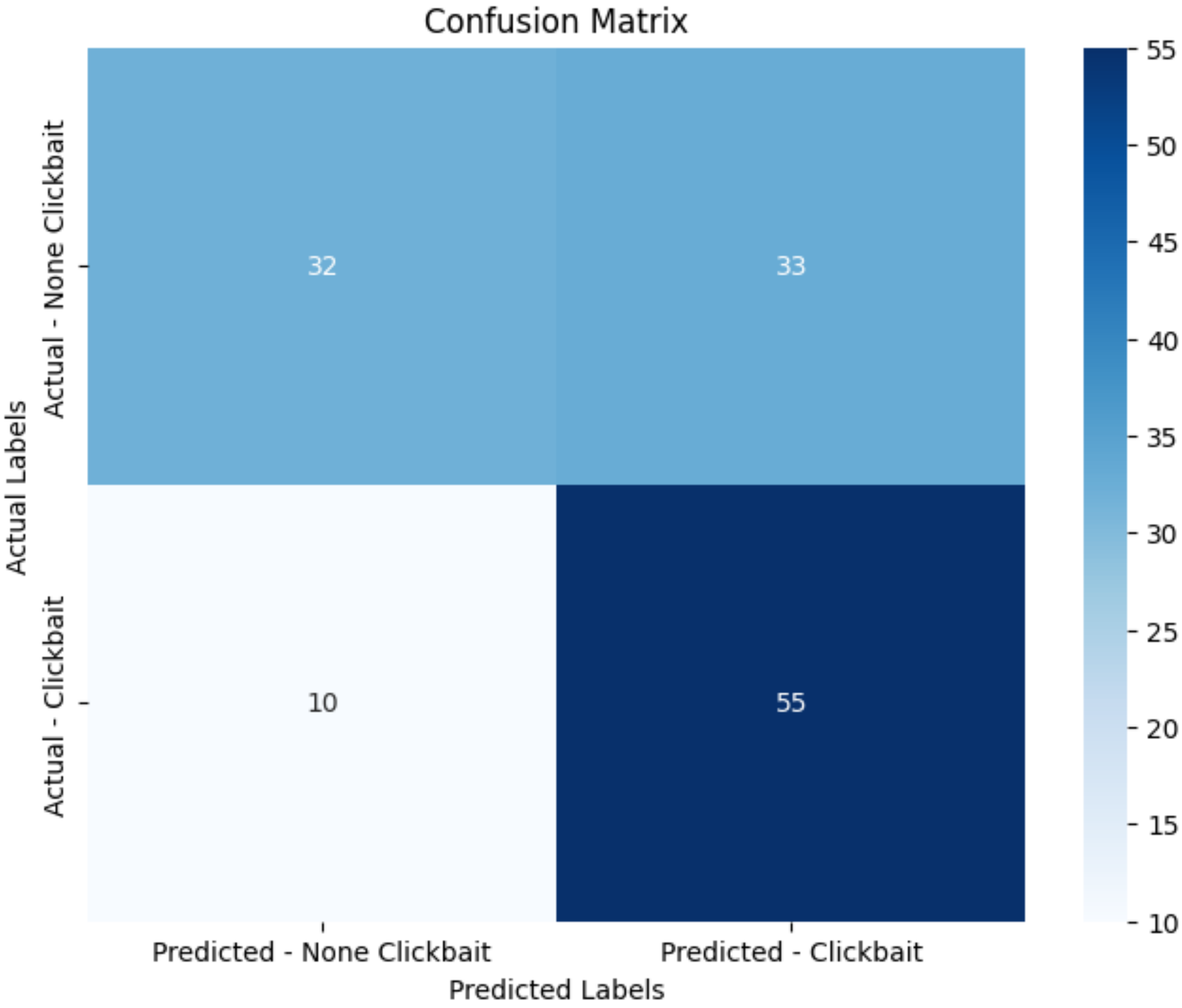


```
              precision    recall  f1-score   support

           0       0.76      0.49      0.60        65
           1       0.62      0.85      0.72        65

    accuracy                           0.67       130
   macro avg       0.69      0.67      0.66       130
weighted avg       0.69      0.67      0.66       130
```

Figure 4.49:Classification Report - Kneighbors Classifier with N- gram range (1,3)Figure

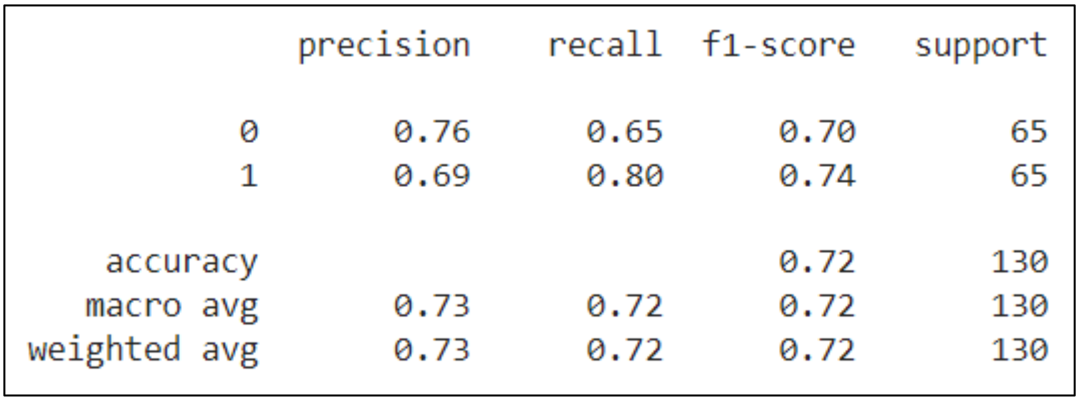


4.50: Confusion Matrix - Kneighbors Classifier with N- gram range (1,3)

4.3 Evaluating the Classifier Performance with Word2Vec Word Embedding Technique

4.3.1 Logistic Regression with Word2Vec word embedding technique

Figures 4.51 and 4.52 present the classification report and confusion matrix for the Logistic Regression model with Word2Vec word embedding technique.



```
              precision    recall  f1-score   support

           0       0.76      0.65      0.70        65
           1       0.69      0.80      0.74        65

    accuracy                           0.72       130
   macro avg       0.73      0.72      0.72       130
weighted avg       0.73      0.72      0.72       130
```

Figure 4.51: Classification Report - Logistic Regression with Word2Vec word embedding technique

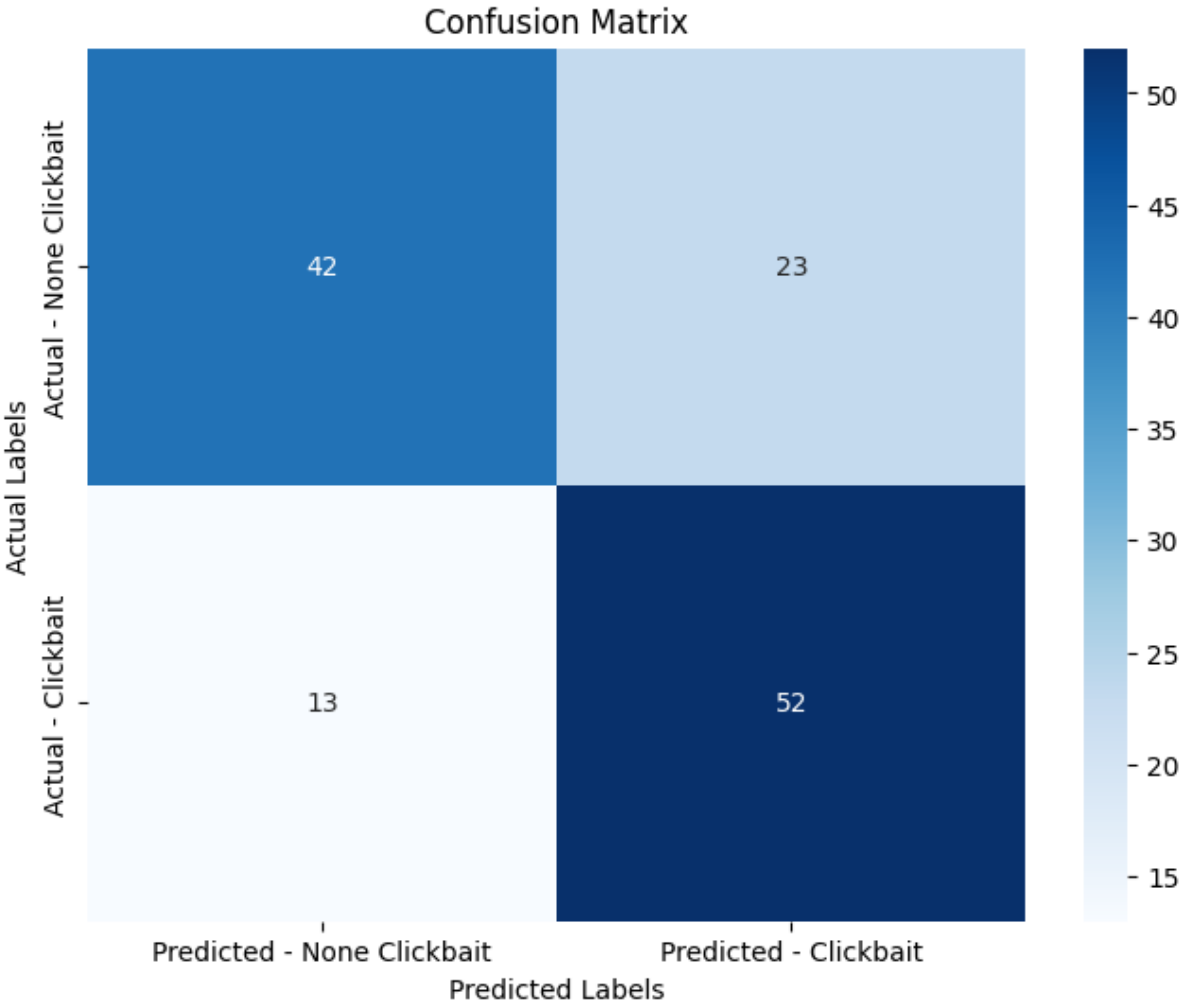
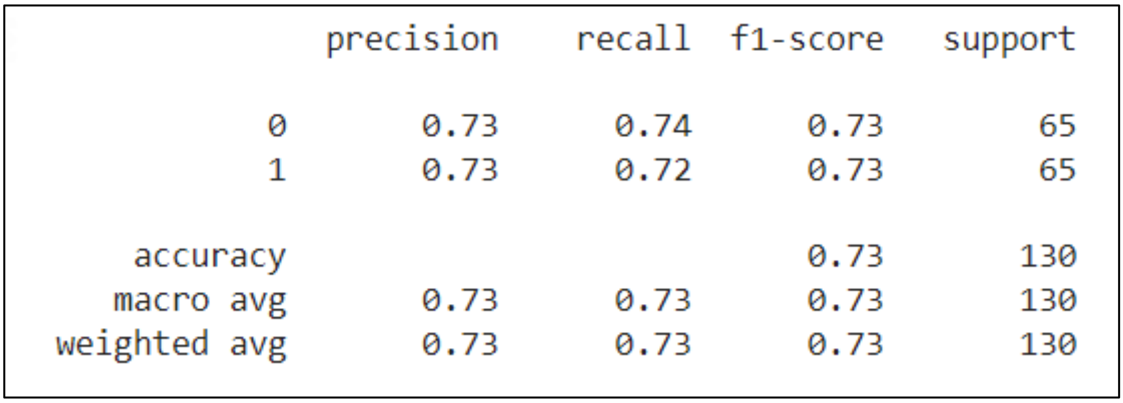


Figure 4.52: Confusion Matrix - Logistic Regression with Word2Vec word embedding technique

4.3.2 Support Vector Classifier with Word2Vec word embedding technique

Figures 4.53 and 4.54 present the classification report and confusion matrix for the Support Vector Classifier model with Word2Vec word embedding technique.



```
              precision    recall  f1-score   support

           0       0.73      0.74      0.73        65
           1       0.73      0.72      0.73        65

    accuracy                           0.73       130
   macro avg       0.73      0.73      0.73       130
weighted avg       0.73      0.73      0.73       130
```

Figure 4.53: Classification Report - Support Vector Classifier with Word2Vec word embedding technique

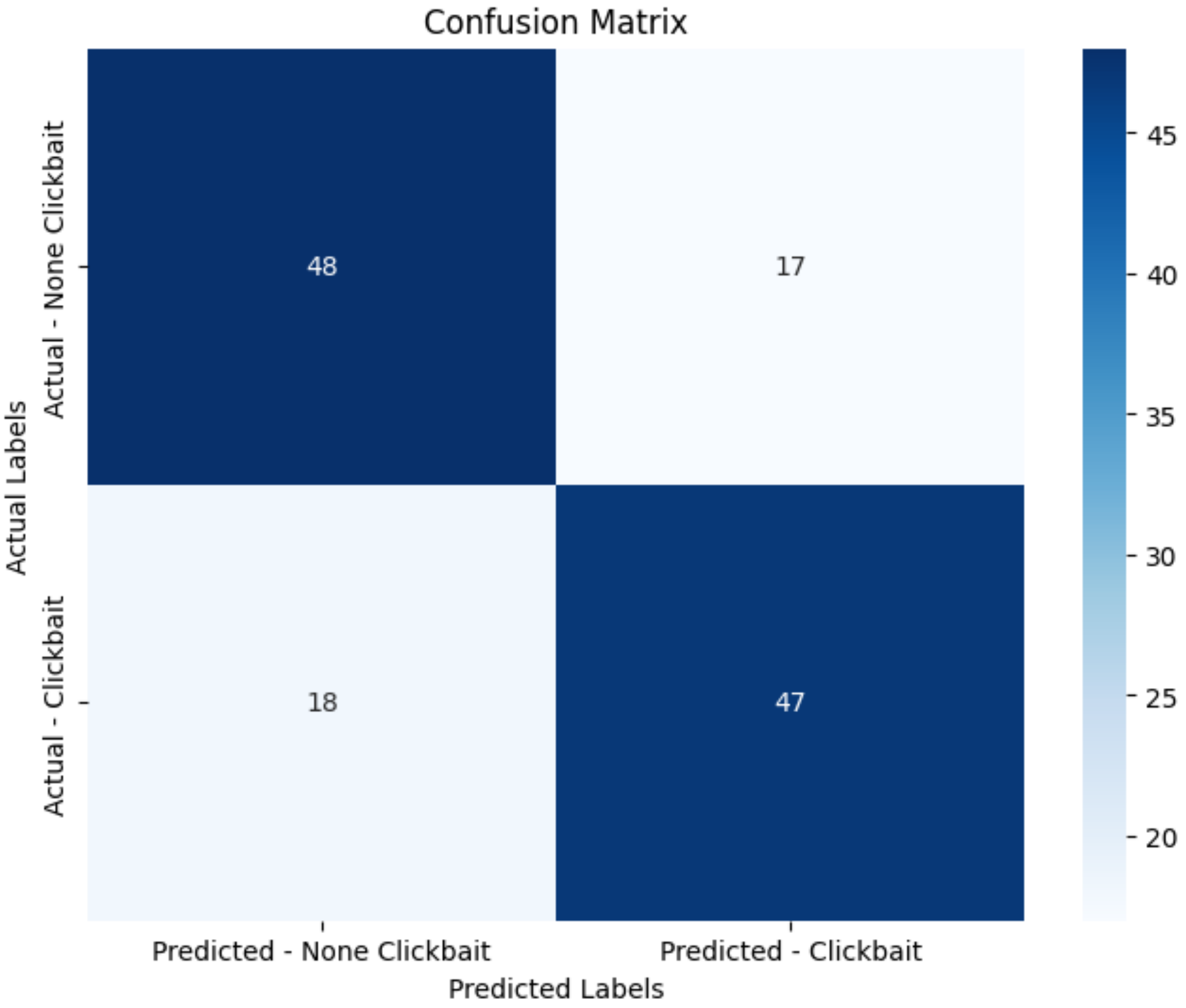
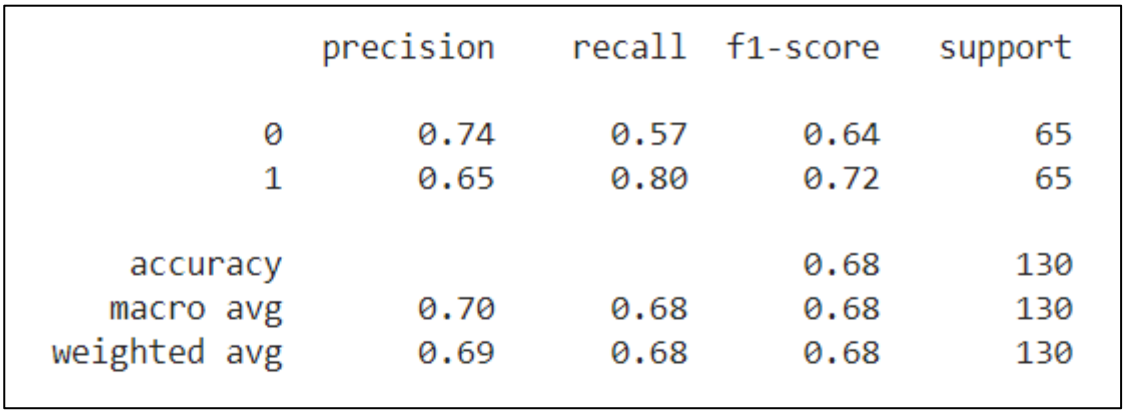


Figure 4.54: Confusion Matrix - Support Vector Classifier with Word2Vec word embedding technique

4.3.3 Kneighbors Classifier with Word2Vec word embedding technique

Figures 4.55 and 4.56 present the classification report and confusion matrix for the Kneighbors Classifier model with Word2Vec word embedding technique.



```
              precision    recall  f1-score   support

           0       0.74      0.57      0.64        65
           1       0.65      0.80      0.72        65

    accuracy                           0.68       130
   macro avg       0.70      0.68      0.68       130
weighted avg       0.69      0.68      0.68       130
```

Figure 4.55: Classification Report - Kneighbors Classifier with Word2Vec word embedding technique

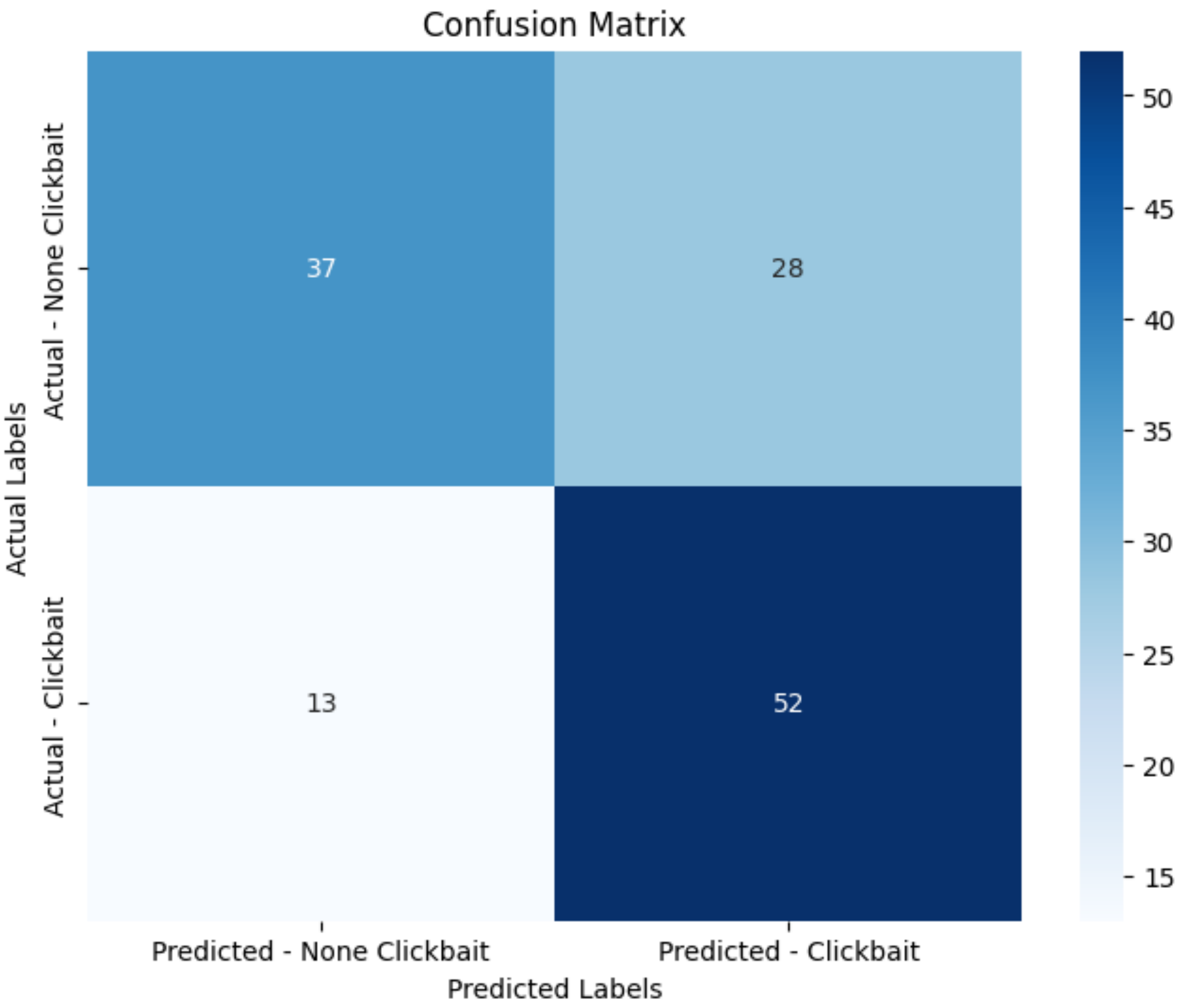


Figure 4.56: Confusion Matrix - Kneighbors Classifier with Word2Vec word embedding technique

4.5 Summary of the Result

4.5.1 Machine Learning Classifiers performance with Count Vectorizer

Table 4.1 shows the different machine learning classifier's performance in different N- Gram ranges with countvectorizer technique.

Table 4.1: Accuracy and F1 Score of Different Machine Learning Classifiers with Count Vectorizer

N-Gram Range	*Machine Learning Model*	*Accuracy*	*F1-Score*
(1,1)	LogisticRegression	0.69	0.68
(1,2)	LogisticRegression	0.66	0.63
(1,3)	LogisticRegression	0.63	0.59
(1,1)	SVC	0.58	0.51
(1,2)	SVC	0.51	0.35
(1,3)	SVC	0.50	0.33
(1,1)	MNB	0.76	0.76
(1,2)	MNB	0.75	0.75
(1,3)	MNB	0.75	0.75
(1,1)	KNN	0.51	0.35
(1,2)	KNN	0.50	0.33
(1,3)	KNN	0.50	0.33

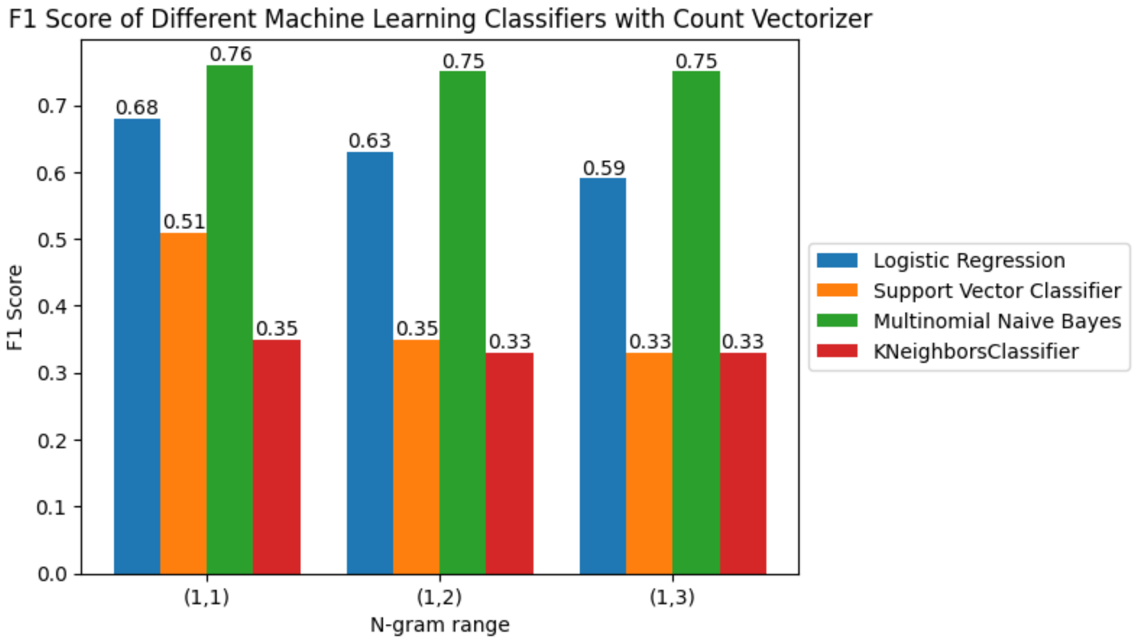


Figure 4.57 : F1 Score of Different Machine Learning Classifiers with Count Vectorizer

According to the observed results, across different N-gram ranges and machine learning models, Multinomial Naive Bayes (MNB) consistently demonstrates strong performance with an accuracy of 0.76 and the F1-score of 0.76 with N-Gram range (1,1). Logistic Regression and Support Vector Classifier (SVC) both exhibit declining performance as the N-gram range increases. In addition, the SVC generally performing slightly worse than Logistic Regression. The K-Nearest Neighbors (KNN) consistently performs the poorest among the models considered, with the F1-score ranging from 0.33 to 0.35 across all N-gram ranges. Overall, MNB emerges as the most reliable model for this classification task.

4.5.2 Machine Learning Classifiers performance with TF-IDF Vectorizer

Table 4.2 shows the different machine learning classifier's performance in different N- Gram ranges with TF-IDF vectorizer technique.

Table 4.2: Accuracy and F1 Score of Different Machine Learning Classifiers with TF-IDF Vectorizer

N-Gram Range	*Machine Learning Model*	*Accuracy*	*F1-Score*
(1,1)	LogisticRegression	0.79	0.79
(1,2)	LogisticRegression	0.81	0.81
(1,3)	LogisticRegression	0.81	0.81
(1,1)	SVC	0.78	0.78
(1,2)	SVC	0.75	0.74
(1,3)	SVC	0.75	0.75
(1,1)	MNB	0.76	0.75
(1,2)	MNB	0.77	0.76
(1,3)	MNB	0.77	0.76
(1,1)	KNeighborsClassifier	0.63	0.60
(1,2)	KNeighborsClassifier	0.66	0.64
(1,3)	KNeighborsClassifier	0.67	0.66

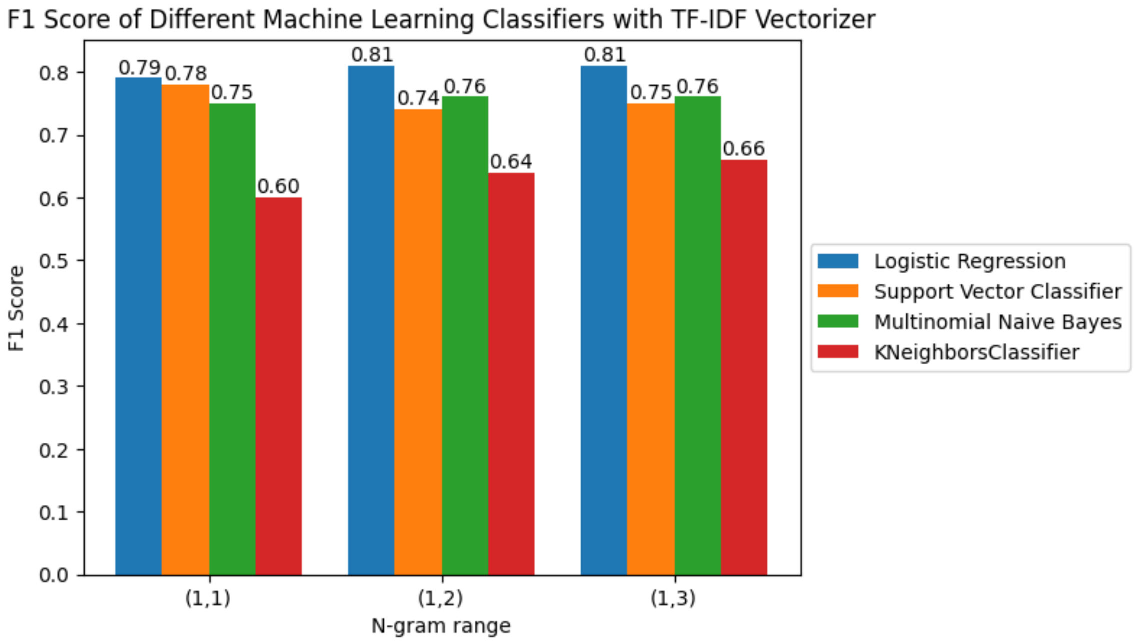


Figure 4.58 : F1 Score of Different Machine Learning Classifiers with TF-IDF Vectorizer

According to the observed results, Logistic Regression consistently demonstrates strong accuracy and F1-scores, showing improvement as the N-gram range increases, peaking at 0.81 for both metrics. Support Vector Classifier (SVC) initially performs well with the F1 score of 0.78 for N-gram range (1,1), but slightly declines to 0.75 for higher N-gram ranges. Multinomial Naive Bayes (MNB) maintains stable performance with F1 score ranging from 0.75 to 0.76 across different N-gram ranges. K-Nearest Neighbors (KNN) shows improvement in F1-score as the N-gram range increases, reaching 0.66. Overall, Logistic Regression emerges as the most consistent performer in this context.

4.5.3 Machine Learning Classifiers Performance with Word2Vec Word Embedding Technique

Table 4.3 and Figure 4.59 shows the Classifiers Performance with Word2Vec Word Embedding Technique.

Table 4.3: Accuracy and F1 Score of Different Machine Learning Classifiers with Word2Vec Word Embedding Technique

Machine Learning Model	*Accuracy*	*F1-Score*
LogisticRegression	0.72	0.72
SVC	0.73	0.73
KNeighborsClassifier	0.68	0.68

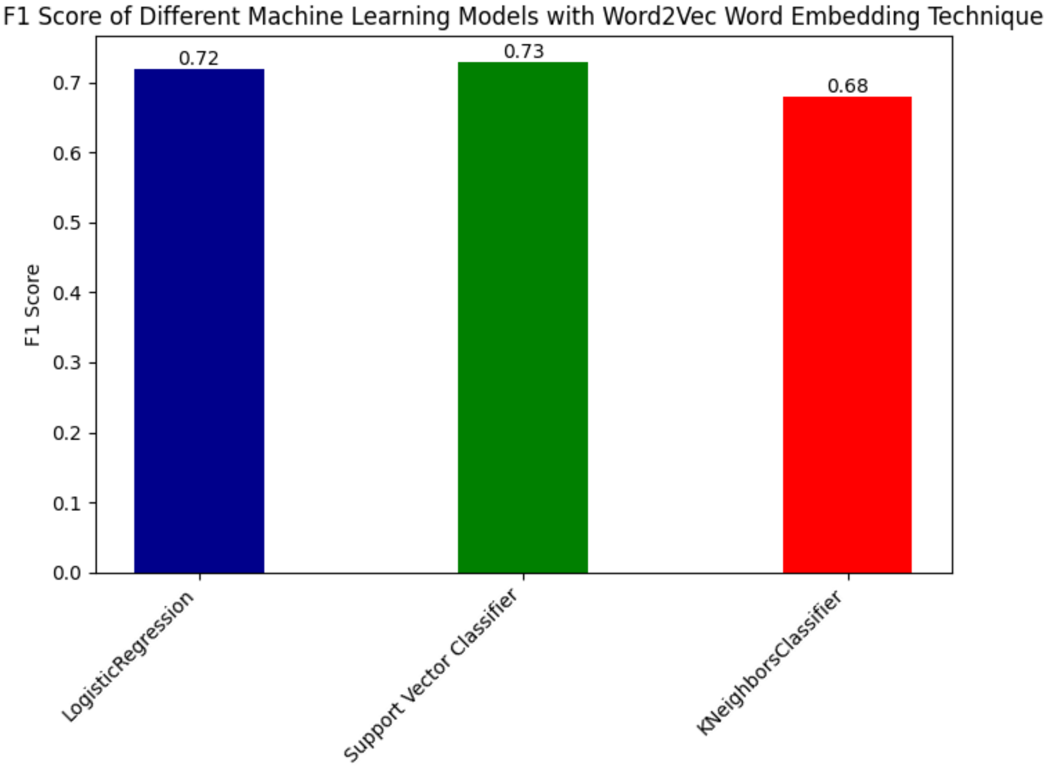


Figure 4.59: F1 Score of Different Machine Learning Models with Word2Vec Word Embedding Technique

In this comparison of machine learning models, Logistic Regression demonstrates a commendable performance, achieving an accuracy and F1-score of 0.72. Support Vector Classifier (SVC) slightly outperforms Logistic Regression, boasting an accuracy and F1-score of 0.73. On the other hand, K-Nearest Neighbors (KNN) falls slightly short of the other models, with an accuracy and F1-score of 0.68. However, as a conclusion Support Vector Classifier can be considered as the best model among the others because of the highest accuracy and F1 score which is 0.73.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this research study, I have observed different approaches to identify clickbait content from the Sinhala YouTube thumbnail. For that I have employed three different text feature extraction techniques which were contvectorizer, TF-IDF vectorizer, and word embeddings, and also employed different ranges of N- Grams including (1,1), (1,2) and (1,3) with respect to the both contvectorizer and TF-IDF vectorizer. Here I have observed the performance of each technique against multiple machine learning algorithms including Logistic Regression, Support Vector Machine, Multinomial Naive Bayes and K-Nearest Neighbors.

Based on the comprehensive analysis of various machine learning classifiers across different vectorization techniques, it is evident that each model exhibits distinct performance characteristics depending on the technique employed.

When utilizing the Count Vectorizer, Multinomial Naive Bayes (MNB) consistently stands out as the top performer, demonstrating robust F1-score which is 0.76 with N-gram range (1,1). Conversely, Logistic Regression and Support Vector Classifier (SVC) show diminishing performance with increasing N-gram ranges.

Transitioning to the TF-IDF Vectorizer, Logistic Regression emerges as the most consistent performer, displaying strongest F1-score which is 0.81 improve with larger N-gram ranges. While SVC initially performs well, its performance diminishes for higher N-gram ranges. MNB maintains stable performance, and KNN also shows improvement as the N-gram range increases.

When employing the Word2Vec word embedding technique, Logistic Regression and SVC exhibit commendable performance, with SVC slightly outperforming Logistic Regression in terms of accuracy and F1-score which is 0.73.

Considering all vectorization techniques and machine learning models evaluated, Logistic Regression with TFIDF vectorization technique performed well with the N gram range (1,2) and (1,3) with the F1 score of 0.81.

As the extended version of this research study, I will be planning to implement a mechanism for collecting feedback from users to continuously improve the model's performance and adapt to trends or variations in clickbait content over time and also planning to build a more comprehensive clickbait identification system by integrating visual features from thumbnails to enhance detection accuracy. Including techniques like object recognition or scene understanding in conjunction with text analysis.

REFERENCES

Arafat, S.Y., Iqbal, M.J., 2020. Urdu-Text Detection and Recognition in Natural Scene Images Using Deep Learning. IEEE Access 8, 96787–96803. https://doi.org/10.1109/ACCESS.2020.2994214

C. P. Chaithanya, N. Manohar, A. B.Issac, 2019. Automatic Text Detection and Classification in Natural Images 7, 5.

Chathuranga, P.D.T., Lorensuhewa, S.A.S., Kalyani, M.A.L., 2019. Sinhala Sentiment Analysis using Corpus based Sentiment Lexicon, in: 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer). Presented at the 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 1–7. https://doi.org/10.1109/ICTer48817.2019.9023671

Davidson, T., Warmsley, D., Macy, M., Weber, I., 2017. Automated Hate Speech Detection and the Problem of Clickbait Language. Proceedings of the International AAAI Conference on Web and Social Media 11, 512–515. https://doi.org/10.1609/icwsm.v11i1.14955

De Saa, E., Ranathunga, L., 2020. Self-Reflective and Introspective Feature Model for Hate Content Detection in Sinhala YouTube Videos, in: 2020 From Innovation to Impact (FITI). Presented at the 2020 From Innovation to Impact (FITI), pp. 1–6. https://doi.org/10.1109/FITI52050.2020.9424875

De Silva, H., Ahangama, S., Perera, S., 2021. Sinhala Text Extraction from YouTube Thumbnails using Convolutional Spiking Neural Networks, in: 2021 From Innovation To Impact (FITI). Presented at the 2021 From Innovation To Impact (FITI), pp. 1–6. https://doi.org/10.1109/FITI54902.2021.9833041

Fatima, S.A., Zafar, A., Malik, K.M., 2023. YouFake: A Novel Multi-Modal Dataset for Fake News Classification, in: 2023 3rd International Conference on Artificial Intelligence (ICAI). Presented at the 2023 3rd International Conference on Artificial Intelligence (ICAI), pp. 148–152. https://doi.org/10.1109/ICAI58407.2023.10136667

Fernando, W.S.S., Weerasinghe, R., Bandara, E.R.A.D., 2022. Sinhala Hate Speech Detection in Social Media Using Machine Learning and Deep Learning, in: 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer). Presented at the 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 166–171. https://doi.org/10.1109/ICTer58063.2022.10024082

Gamage, B., Labib, A., Joomun, A., Lim, C.H., Wong, K., 2021. Baitradar: A Multi-Model Clickbait Detection Algorithm Using Deep Learning, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2665–2669. https://doi.org/10.1109/ICASSP39728.2021.9414424

Gamage, K., Welgama, V., Weerasinghe, R., 2022. Improving Sinhala Hate Speech Detection Using Deep Learning, in: 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer). Presented at the 2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 045–050. https://doi.org/10.1109/ICTer58063.2022.10024103

Garg, S., Gupta, K.K., Prabhakar, N., Garg, A.R., Trivedi, A., 2018. Optical Character Recognition using Artificial Intelligence. International Journal of Computer Applications 179, 14–20.

Jayasuriya, P., Kumarasinghe, B., Ekanayake, S., Munasinghe, R., Thelijjagoda, S., Weerasinghe, I., 2020. Sentiment classification of Sinhala content in social media. https://doi.org/10.1109/SCSE49731.2020.9313023

Medagoda, N., 2016. Sentiment Analysis on Morphologically Rich Languages: An Artificial Neural Network (ANN) Approach. pp. 377–393. https://doi.org/10.1007/978-3-319-28495-8

Medagoda, N., Shanmuganathan, S., Whalley, J., 2015. Sentiment lexicon construction using SentiWordNet 3.0, in: 2015 11th International Conference on Natural Computation (ICNC). Presented at the 2015 11th International Conference on Natural Computation (ICNC), pp. 802–807. https://doi.org/10.1109/ICNC.2015.7378094

Nadarajan, A.S., 2018. A Survey on Text Detection in Natural Images 6.

Pise, A., Ruikar, Dr.S., 2014. Text detection and recognition in natural scene images. Presented at the International Conference on Communication and Signal Processing, ICCSP 2014 - Proceedings, pp. 1068–1072. https://doi.org/10.1109/ICCSP.2014.6950011

Ruwandika, N.D.T., Weerasinghe, A.R., 2018. Identification of Hate Speech in Social Media, in: 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer). Presented at the 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 273–278. https://doi.org/10.1109/ICTER.2018.8615517

Sandaruwan, H.M.S.T., Lorensuhewa, S. a. S., Kalyani, M. a. L., 2020. Identification of Abusive Sinhala Comments in Social Media using Text Mining and Machine Learning Techniques. International Journal on Advances in ICT for Emerging Regions (ICTer) 13, 13–25. https://doi.org/10.4038/icter.v13i1.7213

Sandaruwan, H.M.S.T., Lorensuhewa, S.A.S., Kalyani, M.A.L., 2019. Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning. 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer) 1–8. https://doi.org/10.1109/ICTer48817.2019.9023655

Senevirathne, L., Demotte, P., Karunanayake, B., Munasinghe, U., Ranathunga, S., 2020. Sentiment Analysis for Sinhala Language using Deep Learning Techniques. https://doi.org/10.48550/arXiv.2011.07280

Vitadhani, A., Ramli, K., Dewi Purnamasari, P., 2021. Detection of Clickbait Thumbnails on YouTube Using Tesseract-OCR, Face Recognition, and Text Alteration, in: 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). Presented at the 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST), pp. 56–61. https://doi.org/10.1109/ICAICST53116.2021.9497811