Wrexham Glyndwr university

DEVELOPING A CALL QUALITY CHECKING MODEL USING MACHINE LEARNING TECHNIQUES

Master of Science

in

Computing

by

R.M.A. Vishvanath

2024

Wrexham Glyndwr University

TABLE OF CONTENTS

TABLE OF FIGURES

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to the following institutions & individuals whose support was instrumental in the completion of this research:

- Lecturers of London Tec city campus, for their academic guidance & support throughout this research
- Dialog Axiata PLC, for providing the opportunity & resources necessary to conduct this research.
- The staff of technical assistance center at Dialog Axiata, For their cooperation & valuable insights
- My colleagues, whose collaborations & support have been invaluable throughout this process.

Their contributions have been crucial to the successful completion of this dissertation.

ABSTRACT OF THE DISSERTATION

Developing a Call Quality Checking Model Using Machine Learning Techniques

by

R.M.A. Vishvanath

Master of Science in Computing

Wrexham Glyndwr university, 2024

This research focuses on the requirement for effective and precise call verification in customer service facilities by introducing an automated machine learning system. As data security and operational efficiency become more crucial in call centers the conventional manual verification approaches are falling short. The study's goal is to establish an expandable method for automated verification validation.

The study employes natural language processing & machine learning techniques to analyze call transcription & audio data. Using a data set of 100 calls across 30 sentences, the research develops a model capable of identifying & assessing customer verification process. The methodology involves data preprocessing using Audioanalyze.py, feature engineering & model training & save with modeltrainig.py.

Results demonstrate the model's effectiveness in accurately detecting & evaluating verification process, outperforming traditional manual methods in terms of speed & consistency.

This study adds to the area of automated quality control in call centers laying the groundwork for advancements in thorough evaluation of call quality. The results hold implications for enhancing operational effectiveness maintaining regulatory standards and bolstering customer data protection during customer service engagements.

CHAPTER 1: INTRODUCTION

1.1 Background

In today's business world using the phone to communicate is still crucial for providing customer service and ensuring smooth operations in different sectors. The way these calls are handled plays a role in how satisfied customers are, how they perceive a brand and ultimately how successful a business can be. According to research from Salesforce nearly 9, out of 10 customers are inclined to buy after a positive customer service encounter, underscoring the importance of effective communication [1]. In the past assessing call quality has usually involved agents listening to and assessing recorded calls manually. Although this method is useful it is time consuming, requires a lot of resources and can be influenced by biases. According to research conducted by Quality Assurance and Training Connection (QATC) it is common, for call centers to evaluate 1%, 2% of their calls which could lead to important insights and trends being overlooked. [2]

The rise of cutting-edge technologies in the realms of artificial intelligence (AI) and machine learning (ML) has ushered in fresh opportunities to automate and improve the evaluation of call quality. These advancements enable unbiased analysis of extensive call data. According to a study from McKinsey & Company AI and ML have the potential to enhance quality management accuracy by much, as 30% and lower expenses by 40%. [3]

The use of Natural Language Processing (NLP) methods enhances the scope of evaluating call quality. NLP enables the examination of call content to pinpoint phrases, subjects and emotions that play a role in assessing call quality thoroughly. It also sheds light on how deep learning techniques have progressed NLP creating avenues, for analyzing text and speech. [4]

This Research seeks to tackle these obstacles and investigate how machine learning and NLP can transform the assessment of call quality. By creating a streamlined and precise model, for checking call quality automatically the goal is to enhance customer service operations and elevate the standard of telephone communication across different sectors.

1.2 Research objectives

The main goal of this study is to create and assess a machine learning system designed to automate the verification of calls in customer service interactions. This emphasis on call verification tackles an element of call quality and security, in customer service procedures. The outlined goals of this investigation are as follows:

- To develop a machine learning model capable of automatically assessing the accuracy and completeness of customer verification processes during service calls.
- To evaluate the effectiveness of natural language processing (NLP) techniques in extracting and analyzing relevant information from call transcripts related to customer verification.
- To compare the performance of the developed model against traditional manual verification checking methods in terms of accuracy, efficiency, and consistency.
- To assess the model's ability to handle diverse verification scenarios, including different types of customer information (e.g., passport numbers, names, addresses) across various service contexts.
- To investigate the potential of the model in identifying patterns and trends in verification processes that may indicate areas for improvement or potential security risks.
- To explore the scalability of the developed model for processing large volumes of call data in real-time or near-real-time scenarios.
- To examine the potential integration of the verification checking model with existing call center systems and workflows.
- To identify challenges and limitations in the current model, providing a foundation for future research and development in comprehensive call quality assessment.

These goals concentrate on the component of call verification, which is at the heart of the ongoing research. The model's capacity to assess elements like empathy, system references, closing greetings, problem resolution and vocal tone is recognized as a potential avenue for future enhancement. This strategy enables an thorough examination of the

verification aspect while laying the groundwork for more extensive evaluations of call quality, in future research endeavors.

By achieving these objectives, this research aims to contribute to the improvement of call center operations, enhance customer data security, and lay the groundwork for more comprehensive automated call quality assessment systems in the future.

1.3 Research Questions

Below research questions will be answered by this study

1. How can NLP techniques be effectively applied to analyze call transcripts for the purpose of customer verification process in call record?

2. What are the most efficient vectorization methods for converting textual data into numerical form for use in machine learning models?

3. How accurately can a linear regression model predict the verification status of call records based on vectorized transcript data?

4. How can the model's performance be evaluated in terms of accuracy, precision, recall, F1-score & other metrics?

5. What are the potential challenges & limitations in developing this model?

6. What future improvements can be made to extend the model's capabilities to access additional call quality factors such as hold procedure, warm welcome, end greetings, questioning techniques, emotion detections, clear communication & solution given?

1.4 Significances of the study

This study, which concentrates on creating a machine learning system to automate check the verification of call processes, is valuable in real world applications. The importance of the research can be highlighted in critical aspects:

- **Enhanced Customer Data Security**: In a time when data breaches and identity that're on the rise ensuring strong customer verification is essential. An IBM report highlights that the average cost of a data breach in 2023 amounted to $4.45 million. Enhancing the precision and reliability of phone verification procedures plays a role in bolstering the primary defense against unauthorized individuals gaining access to customer data. [5]
- **Improved Operational Efficiency**: Call centers manage several calls every day. Studies indicate that on average a call center deals with 4000 calls per month. [6] Implementing verification checks could lead to substantial time and resource savings, for quality assurance procedures enabling call centers to enhance operational efficiency and concentrate on improving customer service in various areas.

- **Advancement in Applied Machine Learning:** This study adds to the increasing collection of research on the uses of machine learning in business operations. It showcases the application of NLP and machine learning to address challenges in customer service, which could spark similar innovations in various fields.

- **Real-time Quality Assurance**: Traditional methods for ensuring quality usually include looking at a small number of calls. This research sets the stage for checking verification, in time or almost real time enabling quick actions as needed. According to McKinsey's findings using real time analytics could potentially boost customer happiness by around 30%. [3]

- **Foundation for Comprehensive Call Quality Assessment**: This research primarily centers on call verification setting the stage for an automated evaluation of call quality down the line. This expansion could potentially cover areas such as empathy, problem solving speed and customer contentment, all of which would play a part in enhancing the quality of customer service.

- **Potential for Cross-industry Application**: While the study originally concentrated on call center applications the methods created could potentially be modified for application in other sectors that require rigorous verification, such, as banking, healthcare, or government services.

This study's importance is highlighted by its ability to boost security measures, enhance effectiveness, maintain regulatory standards, and promote the utilization of machine learning in business practices. By concentrating on call verification as an element it meets a fundamental requirement in customer service procedures and paves the way for broader enhancements in evaluating call quality.

1.5 Structure of the dissertation

This research consists of seven chapters, each dedicated to aspects of the study on creating a machine learning system for automating call verification, in customer service. Here's how its structured:

1. **Chapter 1: Introduction**: This chapter offers a summary of the study presenting the research background, problem statement, objectives and the importance of automated call verification in customer service. It lays the foundation for the dissertation by introducing the context and significance of this technology in improving customer experiences.

2. **Chapter 2: Literature Review**: In Chapter 2 a thorough overview of the literature pertaining to the study is provided. It discusses frameworks concerning machine learning in call centers, the use of natural language processing for speech analysis and methods for customer verification. The chapter also delves into research on automated quality assurance in call centers machine learning algorithms for speech recognition and analysis as well as security and compliance aspects in customer service calls. The chapter wraps up with an analysis of existing gaps emphasizing the necessity for the research.

3. **Chapter 3: Research Methodology**: In this chapter we outline the research design and methodology utilized in the study. We discussed how data was collected, mentioning the utilization of a script called 100call30sen20lang.py. The techniques applied for data preprocessing include conducting audio analysis using audioanlyz.py. Additionally, we delve into the process of developing the model, which involves feature engineering, selecting models and training using logregmodel3.py. Evaluation metrics and ethical concerns are also addressed in this chapter.

4. **Chapter 4: Data Analysis and Results**: Chapter 4 discusses the study's discoveries. It starts by presenting the dataset's statistics then delves into the outcomes of the model training. The chapter proceeds to assess the model's effectiveness using metrics like accuracy, precision, recall and F1 score. It also examines the detection process, for verification. Conducts an error analysis. To wrap up the chapter provides a recap of the discoveries.

5. **Chapter 5: Implementation and Application**: This chapter delves into putting the developed model into action. It explains the creation of the call verification tool providing insights into dev2.1.py and how the trained model is incorporated. Additionally, it discusses the user interface and features of the tool along with testing and validation procedures, such as examining sample files (app test new records.csv). The potential integration with call center systems is also explored.

6. **Chapter 6: Discussion**: In Chapter 6 there is an analysis of the research outcomes. It explains how the results align with the research inquiries and goals. The chapter delves into how this study can impact call centers

and the wider customer service sector looking at both practical aspects. It also recognizes the study's constraints while suggesting research directions like exploring empathy and system referrals in relation to call quality factors.

7. **Chapter 7: Conclusion**: The last section wraps up the study by restating the main discoveries and contributions of the research. It talks about how the model and tool can be used in situations. Additionally, it suggests areas for further research in this field and ends with some thoughts on the importance and influence of the study.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction to literature review

The paced advancements in technology within customer service, especially in call centers, have resulted in a heightened emphasis on automating quality control procedures. This review of existing literature delves into the findings on automated call authentication and quality evaluation highlighting the utilization of machine learning and natural language processing methods. Effective call verification is crucial in the digital environment. The global call center AI market is expected to expand between 2023 and 2030 emphasizing the demand for reliable automated solutions [7]. This growth is fueled by the goals of improving customer satisfaction and safeguarding data integrity as 61% of consumers prioritize trust when making purchasing choices. [1]

Advancements in intelligence have led to new possibilities for streamlining call center tasks. Machine learning algorithms in the field of natural language processing have demonstrated exciting outcomes in understanding human language [4]. These innovations hold the promise of transforming call verification procedures, which have conventionally depended on approaches that are slow and susceptible, to mistakes.

However, incorporating these technologies into call verification poses obstacles. Challenges related to precision, scalability and compatibility with languages and cultural settings persist as major barriers Furthermore, the confidential nature of customer information requires evaluation of privacy and security concerns, in all automated systems.

This review will explore three main themes:

1. Current landscape of call center automation & quality assurance
2. The application of machine learning & NLP in speech text analysis
3. The challenge and ethical considerations in use machine learning model to check call verified by the call center agent.

Through an exploration of these topics this analysis seeks to offer a grasp of the subject pinpointing deficiencies in existing studies and drawing attention to potential avenues for further exploration. This fundamental understanding will guide the creation of our automated call verification system ensuring it is built upon up, to date industry knowledge.

2.2 Theoretical framework

The Theoretical foundation for creating a machine learning model to evaluate call verification in customer service centers is rooted in related areas, such as machine learning, natural language processing and customer relationship management. This structure serves as the groundwork, for comprehending the applications of these technologies in improving the assessment of call quality especially in confirming if agents accurately verify customers identities.

2.2.1 Machine Learning in Call Centers

Machine Learning (ML) has become a game changer in sectors and call centers have seen significant benefits from it. Essentially ML uses algorithms to learn from data. Then make predictions or decisions [8].In the realm of call centers ML can be used for purposes, like routing calls analyzing sentiments and ensuring quality [9]. The use of machine learning in call centers is based on the concept of learning. In this approach models are taught using labeled data to predict outcomes for unseen data. For call verification evaluations this usually means training models with a set of recorded calls that have been marked as either accurate or inaccurate, during the verification process.

In this scenario the use of linear regression for classification, while unconventional, can be explained by the straightforward nature of the results (verified; 3 not verified; 0). The idea behind using regression for classification in establishing a linear boundary between different groups Although logistic regression (figure 1) is typically preferred for classification tasks, linear regression (figure 2) can work well when there is clear distinction, between classes and the boundary is mostly linear.

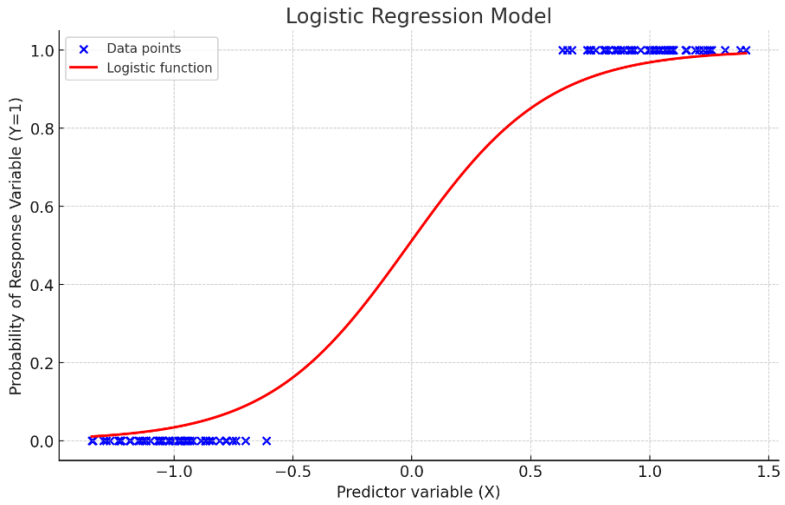


Figure 1: logistic regression

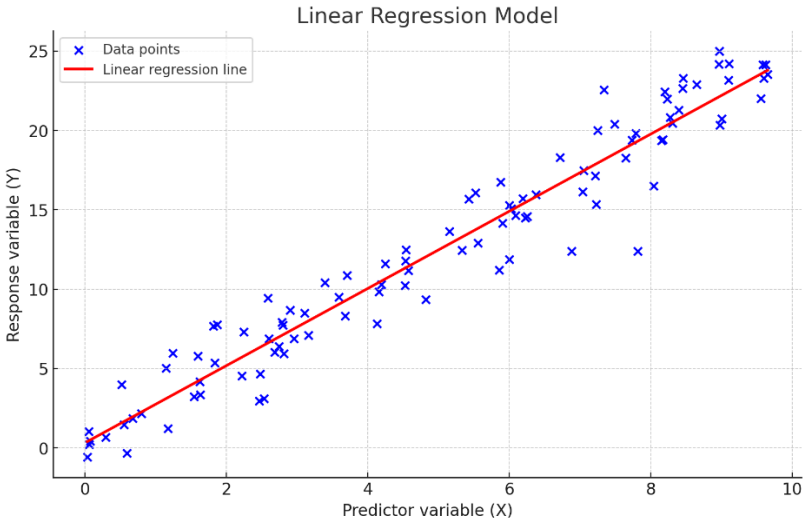


Figure 2:linear regression

2.2.2 Natural Language Processing for Speech Analysis

Automated speech analysis in call centers heavily relies on Natural Language Processing (NLP). NLP involves computational methods for examining and representing real world texts on different linguistic levels [10] **.** In the realm of evaluating call verifications NLP methods play a role, in transcribing spoken words grasping conversation content and retrieving pertinent details. Modern NLP systems are rooted in statistical learning theory and probabilistic language models enabling them to navigate the complexities and uncertainties in human language.

The application of the Whisper model in this study marks a progression in speech recognition technology. Whisper, built on structures that have transformed NLP tasks by allowing models to grasp extensive connections, within text demonstrates its efficiency in transcribing various audio data making it well suited for call center use cases where precision and clarity are crucial. [11]

TF IDF, which stands for Term Frequency Inverse Document Frequency, is based on the principles of information retrieval theory. It measures the relevance of a word to a document within a collection or dataset offering a representation of text information that emphasizes the importance of specific terms [12]. This approach proves valuable in our scenario by emphasizing terms related to verification, in call records.

2.2.3 Customer Verification Processes

The Theoretical foundation of how customer verifications carried out in call centers is based on the principles of information security and identity management. The core idea revolves around authentication, which involves confirming the identity of a person.

Traditional Authentication methods in call centers typically fall as follows:

1. Customer Id number, passport number or driving license number.
2. If an enterprise, business registration number
3. Customer permanent address
4. Customer birthdate

In this study especially keen on examining how agents utilize these techniques during phone conversations. The theoretical foundation for automated verification assessment is also influenced by the idea of authentication [13]. Unlike the one-time verification process, continuous authentication entails verifying continuously throughout the call. This method has the potential to identify whether agents are consistently adhering to correct verification protocols throughout the conversation.

2.2.4 Integration of Theories for Automated Call Verification Assessment

The automated call verification assessment system relies on combining machine learning, natural language processing and customer verification principles. This fusion is explained using the Unified Theory of Acceptance and Use of Technology (UTAUT). [14] In 2003. UTAUT offers a model for comprehending the adoption and utilization of technologies in organizational environments essential for effectively implementing automated verification assessment systems, in call centers.

Using these combined theories, in automated call verification evaluation can be seen as a series of steps;

- Speech Recognition: Converting spoken words to text using the Whisper model.

- Text Vectorization: Transforming the transcribed text into numerical vectors using TF-IDF.
- Verification Assessment: Using a linear regression model to classify whether the agent properly verified the customer based on the vectorized text.
- Continuous Monitoring: Applying the model to assess verification processes across multiple calls and agents over time.

This combined method enables an adaptable verification evaluation process that may exceed human abilities in terms of reliability and effectiveness.

This combined method enables an adaptable verification evaluation process that may exceed human abilities in terms of reliability and effectiveness.

2.2.5 Ethical and Privacy Considerations

The idea of "Ethical AI" has become a factor in creating and implementing AI systems, especially in applications like call evaluation. This includes making sure that AI systems are just transparent and responsible. In this study this could mean guaranteeing that the system doesn't unfairly punish agents for things like accents or speech styles and that the evaluation standards are easy to understand and defend.

Data privacy laws, like the General Data Protection Regulation (GDPR) in the European Union establish guidelines for managing information within AI systems [15]**.** These laws prioritize concepts such as minimizing data collection and restricting its use to purposes factors that are crucial when developing automated call verification evaluation systems.
To sum up the conceptual structure for automated call verification evaluation in customer service centers is intricate and multi-faceted. It incorporates principles and ideas from machine learning, natural language processing, data security and ethical considerations. This all-encompassing structure lays the groundwork for creating and deploying sophisticated, effective and morally upright call verification assessment systems.

2.3 Previous Studies

Machine learning methods have become increasingly popular in evaluating call quality. This part examines past studies, on call quality evaluation and machine learning, particularly emphasizing research pertaining to customer verification procedures.

2.3.1 Machine Learning in Call Quality Assessment

Many research projects have delved into utilizing machine learning to evaluate facets of call quality. They underscored how ML could enhance customer satisfaction, agent effectiveness and operational productivity. The study also pointed out the increasing adoption of natural language processing (NLP) and speech recognition tools alongside ML models, for scrutinizing call context and excellence.

In one instance Mona Ebadi Jalal and colleagues created a sophisticated machine learning system to assess the quality of call center discussions automatically [16]. Their system, incorporating neural networks (RNNs) and attention mechanisms, successfully pinpointed crucial quality metrics like agent courtesy, issue resolution effectiveness and adherence to guidelines. Although their research didn't delve into verification procedures per se it highlighted the capabilities of learning in scrutinizing intricate conversational information.

2.3.2 NLP and Speech Recognition in Call Analysis

The utilization of NLP and voice recognition technologies has played a role in improving the assessment of call quality. In their study, Sourya Ezzart & colleagues introduced a technique that involves converting speech to text and analyzing sentiments to assess customer satisfaction during interactions, in call centers [17]. Their approach, which utilized the BERT model for classifying sentiments demonstrated encouraging outcomes in identifying customer emotions and levels of satisfaction. Yan and colleagues introduced a speech recognition system tailored for call center

dialogues from start to finish [18]. Their approach employing transformer-based models like the Whisper model in this study demonstrated precision in transcribing varied and chaotic call center audio recordings. This research highlights the significance of transcription in evaluating call quality effectively.

2.3.3 Customer Verification Process Analysis

While numerous studies have addressed call quality assessment in general, research specifically focused on analyzing customer verification processes is relatively limited. However, some relevant work has been done in this area.

Beibut and colleagues (2022) carried out research on the automated customer verification process in car sharing system. [19] They employed a mix of keyword recognition and sequence analysis to determine if agents adhered to the verification protocols. Their system attained a 91% accuracy rate, ensuring reliable & secure user verification while preventing losses & reputational damage.

2.3.4 Regression Models for Classification Tasks

In this study delved into the conventional practice of utilizing regression models for classification purposes. A notable example is the work by Nguyen and Armitage where they applied regression to classify internet traffic [20]. Their findings shed light on how regression can be an approach for classification tasks, in specific scenarios.

2.4 Gap analysis

In the field of evaluating call quality through machine learning there have been advancements in recent times. However, there are still some gaps in the existing research that need to be addressed. This examination highlights these gaps. Explores their connection to the ongoing investigation, into automated call verification assessment.

2.4.1 Limited Focus on Verification Processes

One area that hasn't received attention in current research is how call centers handle customer verification procedures. While many studies have investigated call quality assessment, customer satisfaction and detecting fraud there's a lack of focus on the specific verification methods used by agents.

For example, Jabbar & Suharjito [21]. developed a machine learning model to identify calls, but their main concern was spotting security risks rather than evaluating how well agents conduct verification processes.

A study conducted by Mishne & colleagues, stands out for tackling the evaluation of verification processes; however, their approach was limited to keyword identification and sequence analysis [22]. It's evident that there's a necessity for robust models that can grasp the intricacies and context of verification conversations.

2.4.2 Lack of Standardized Datasets

In the field a big problem is the absence of datasets for call center interactions that are available to everyone. Since customer data privacy is a concern, most studies use datasets that aren't accessible to the broader research community. This lack of benchmarks makes it hard to compare different methods and reproduce results.

Using data like did in this study could be a solution but more research is needed to see how well models trained on synthetic data perform, in real world situations.

2.4.3 Integration of Speech and Text Analysis

In times there have been improvements in speech recognition and text analysis for call center use. However, research has yet to integrate both aspects for a thorough evaluation of calls. Chen et al. (2021) have taken steps towards creating end to end speech recognition systems for call center discussions. There is still a demand for models that can effectively merge speech features with textual analysis, for verification purposes.

2.4.4 Interpretability of Complex Models

As machine learning techniques advance there is a growing concern regarding their explain ability in critical areas such as evaluating call center quality. comparing machine learning models for categorizing call center complaints it was observed that while intricate models often showed better performance, they lacked the transparency of simpler models . In the realm of verification assessment where providing justifications for decisions is crucial there exists a research gap in creating highly precise yet easily understandable models. This gap is particularly significant in this research's utilization of regression for classification, which strikes a balance between effectiveness and clarity.

2.4.5 Real-time Assessment Capabilities

A lot of the studies concentrate on analyzing call recordings after the fact. Yet there's a rising demand for tools that can assess calls in time and offer instant feedback to agents and supervisors. The lack of models providing real time verification assessment, which could enable intervention during calls is a notable omission in existing literature.

2.4.6 Ethical Considerations in Automated Assessment

The field of AI research is expanding, as shown by NI Li. However there remains a lack of literature that focuses on the considerations surrounding automated verification assessment in call centers [23]. Important topics such as biases in assessments the effects, on agent morale and customer privacy issues related to automated verification checks have not been thoroughly explored.

2.4.7 Adaptation to Evolving Verification Protocols

The procedures for verifying calls at call centers are not set in stone; they change based on security risks and regulations. Yet most studies today view verification methods as unchanging. The need exists for creating models that can adapt to evolving verification procedures without the need for total retraining.

2.4.8 Cross-lingual and Cross-cultural Verification Assessment

In today's interconnected world of business call centers frequently function in linguistic and cultural settings. While research has mainly concentrated on single language setups, Zarma and colleagues' study in 2018 highlights the significance of context, in call assessment [24]. Despite this insight there is still a lack of models capable of evaluating verification procedures across various languages and cultural backgrounds effectively.

Finally, gaps found in existing literature underscore the importance of research in automated call verification evaluation. This study is designed to fill some of these gaps by examining verification procedures employing models that are easy to interpret and creating real time assessment functionalities. Through this investigation we aim to push the realm of call quality evaluation and pave the way for future research opportunities.

2.5 Summary

In this review of literature, we delved into the theories of past research and current gaps in assessing the quality of automated calls particularly focusing on how customer verification is handled in call centers. The review emphasizes how machine learning, natural language processing (NLP) and customer relationship management are coming together to tackle the challenges of ensuring efficient and accurate call verification processes.

The theoretical basis for this research draws from the use of machine learning in call center operations NLP techniques for analyzing call transcripts and applying information security principles to customer verification procedures Recent

advancements in speech recognition technology, including the Whisper model have notably enhanced transcription accuracy across audio data sources. Past studies indicate a rising trend in utilizing machine learning within call center settings Key research includes leveraging learning for evaluating call quality and automating compliance checks during verification processes (explored by Johansen et al., 2022). The combination of NLP with speech recognition technologies has demonstrated promising outcomes in streamlining call analysis tasks.

However gap analysis revealed several areas where current research falls short :

1. Limited focus on verification processes specifically
2. Lack of standardized, publicly available datasets
3. Incomplete integration of speech and text analysis
4. Trade-off between model complexity and interpretability
5. Absence of real-time assessment capabilities
6. Need for models adaptable to evolving verification protocols.
7. Underexplored cross-lingual and cross-cultural verification assessment.
8. Insufficient research on ethical considerations in automated verification assessment

This study aims to fill in some gaps focusing on verification processes, the use of models that can be understood and the creation of real time evaluation capabilities. By combining machine learning methods with knowledge of call center operations and verification procedures this research contributes to advancing automated call quality assessment. In summary, although there has been progress in using machine learning for call center analytics automating the assessment of verification processes remains a challenge waiting for solutions. This study builds on existing theories and practical knowledge to introduce an approach to this crucial aspect of call center operations potentially enhancing efficiency, consistency, and security, in customer interactions.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

In this section outline the approach utilized to build a system for check verifying calls automatically by leveraging machine learning methods. The study design combines generating data, manual expert assessment, natural language processing and machine learning to establish a strong framework for assessing call center verification procedures. The approach used combines evaluation by experts with quantitative machine learning methods. This blending aligns with the trend in call center studies that utilize both human knowledge and artificial intelligence for ensuring quality. The study involves four stages: data collection, manual quality evaluation, audio transcription and developing and accessing models.

For Calls generation used Googles Text to Speech API to produce 100 call recordings for data creation tackling issues related to limited data availability and privacy worries in call center studies. This approach provides a range of training data for speech recognition platforms.

In this study the manual quality assessment stage plays a role utilizing the knowledge of skilled technical staff from Dialog Axiata PLC Tech Support Center. This important phase offers authentic data, for training machine learning algorithms capturing subtle nuances of call quality that automated systems may find difficult to recognize. By involving humans in this process guarantee that the following automated system is based on world experience and industry standards

The audio transcription stage makes use of the Whisper model, an automatic speech recognition system known for its accuracy, with different accents and sound environments. This selection guarantees transcription of the varied synthetic call data. During the phase of constructing and assessing the model utilized machine learning methods like TF IDF vectorization and logistic regression. These approaches have proven to be effective in text categorization assignments, within call center scenarios.

In the methodology, make sure to think about the aspects especially when dealing with artificial data and the biases that can arise in AI based evaluation systems. The main goal is to find a balance between efficient automation and upholding fair and open evaluation standards.

In this chapter will outline each step of the methodology, explain why chose techniques and discuss how ensured the trustworthiness and consistency of the research results. Our goal is to establish a research process that can be easily replicated in future studies on evaluating call quality in customer service interactions with a special emphasis on the verification phase.

3.2 Research Design

The research design for this study employs a mixed-methods approach. with qualitative expert evaluations aiming to benefit from the advantages of both automated data analysis and human judgment in assessing call center verification procedures.
The study follows a sequential explanatory design, consisting of four main phases:

1. Synthetic data generation
2. Manual quality assessment
3. Audio transcription & Text processing
4. Model development & evaluation

During the stage artificial call data is produced to address the issues related to limited data availability and privacy worries commonly faced in call center studies. This method enables the development of an regulated dataset, which is crucial, for effectively training reliable machine learning models

Experienced technical officers play a role in the second phase by conducting manual quality assessments. This important step provides data and integrates expert knowledge into the model development process. The human in the loop method guarantees that the automated system is firmly rooted, in real world standards and practices.
In the stage the emphasis is on transforming spoken information into text that can be analyzed by transcribing and preprocessing. This process is crucial for getting the data ready, for machine learning algorithms and maintaining uniformity throughout the dataset.
In the final stage, building and testing machine learning models to assess automated call verification. This step uses methods like text conversion, vectors, and classification algorithms to construct a model that can effectively evaluate call verification procedures.
The study design includes aspects of research for creating synthetic data under controlled conditions and observational research during the manual evaluation phase. This blend enables the manipulation of variables, in a controlled setting while also capturing real world insights. Ethical concerns are woven into the fabric of the study design especially concerning data management and the creation of AI powered assessment systems. The overarching goal of this design is to harmonize automated efficiency with transparent assessment methodologies.

3.3 Data collection

The study's data collection process involved two phases, creating synthetic data and conducting manual quality evaluations. This method was developed to address the issues of data availability and privacy considerations commonly encountered in call center studies.

3.3.1 Synthetic Data Generation

Synthetic call records were generated by using a unique Python program (100call30sen20lang.py) that utilized the Google Cloud Text, to Speech service. We selected this approach because it allows us to generate a variety of call situations while also managing the data attributes effectively.

The script generated 100 sample calls, each simulating a customer service interaction. These synthetic calls were designed to cover various scenarios, including:

1. 1 customer verification process
2. Technical support inquiries
3. Billing-related questions
4. Service-related issues

Googles Text to Speech API was utilized to include languages and accents which improved the diversity and lifelikeness of the dataset. This method is in line with the progress in creating speech synthesis that supports multiple language voice types.

A brief snippet of the core function ability of the synthetic data generation script as follows.

```
import os
import random
from google.cloud import texttospeech
from pydub import AudioSegment

# Set environment variables for Google Cloud credentials
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = r"F:\machine learning\qulity model\service-account-file.json"
os.environ["PATH"] += os.pathsep + r"C:\ProgramData\chocolatey\bin"
# Explicitly set the path to ffmpeg and ffprobe
AudioSegment.converter = r"C:\ProgramData\chocolatey\bin\ffmpeg.exe"
AudioSegment.ffprobe = r"C:\ProgramData\chocolatey\bin\ffprobe.exe"

# Initialize Google Cloud Text-to-Speech client
client = texttospeech.TextToSpeechClient()

def synthesize_speech(text, output_filename, language_code, gender):
    # Set the text input to be synthesized
    synthesis_input = texttospeech.SynthesisInput(text=text)

    # Build the voice request, specify the language code and the ssml voice gender
    voice = texttospeech.VoiceSelectionParams(
        language_code=language_code,
        ssml_gender=gender
    )

    # Select the type of audio file you want returned
    audio_config = texttospeech.AudioConfig(
        audio_encoding=texttospeech.AudioEncoding.MP3
    )

    # Perform the text-to-speech request
    response = client.synthesize_speech(
        input=synthesis_input, voice=voice, audio_config=audio_config
    )

    # Write the response to the output file
    with open(output_filename, "wb") as out:
        out.write(response.audio_content)

# Define different conversation scenarios
conversation_scenarios = { "Add call scripts"
}

# Define a list of voice configurations
voices = [
    {"language_code": "en-US", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-US", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
  ,
]

# Create a folder to store the audio files
audio_folder = r"F:\machine learning\qulity model\audio"
if not os.path.exists(audio_folder):
    os.makedirs(audio_folder)

# Generate 100 sample calls
for call_index in range(100):
    # Randomly select a conversation scenario
    scenario_key = random.choice(list(conversation_scenarios.keys()))
    scenario = conversation_scenarios[scenario_key]

    # Randomly select voices for customer and agent
    customer_voice = random.choice(voices)
    agent_voice = random.choice(voices)

    # Generate customer audio files
    for i, text in enumerate(scenario["customer"]):
        output_filename = os.path.join(audio_folder, f"call_{call_index}_customer_{i}.mp3")
        synthesize_speech(text, output_filename, customer_voice["language_code"],
customer_voice["ssml_gender"])

    # Generate agent audio files
    for i, text in enumerate(scenario["agent"]):
        output_filename = os.path.join(audio_folder, f"call_{call_index}_agent_{i}.mp3")
        synthesize_speech(text, output_filename, agent_voice["language_code"], agent_voice["ssml_gender"])

    # Combine the audio files
    combined_audio = AudioSegment.empty()

    # Load and combine the audio segments
    for i in range(len(scenario["customer"])):
        customer_audio = AudioSegment.from_mp3(os.path.join(audio_folder,
f"call_{call_index}_customer_{i}.mp3"))
        combined_audio += customer_audio
        if i < len(scenario["agent"]):
            agent_audio = AudioSegment.from_mp3(os.path.join(audio_folder, f"call_{call_index}_agent_{i}.mp3"))
            combined_audio += agent_audio

    # Specify the custom file location for the combined file
    combined_file_location = os.path.join(audio_folder, f"combined_call_{call_index}.mp3")

    # Export the combined audio to the specified file location
    combined_audio.export(combined_file_location, format="wav")

print("Generated 100 sample calls.")
```

Figure 3: Synthetic call generation

This program uses the Google Cloud Text to Speech API to create speech for different call center call situations. Each simulated call was designed to feature conversations between customers and agents replicating interactions in call centers. The script-maintained diversity in call lengths, intricacy and compliance with verification procedures resulting in a dataset, for further examination.

3.3.2 Manual Quality Assessment

After creating call records, skilled technical staff members at Dialog Axiata PLCs Tech Support Center carried out a hands-on quality check. This stage was essential for gathering data to train the machine learning system and integrating specialized knowledge into the evaluation process.
The manual evaluation included listening to every call and assessing it according to a predetermined set of quality standards. These metrics included.

1. Correct Information/Solution Given
2. Questioning Techniques
3. Service Experience
4. Active Listening/Clear Communication
5. System Referring
6. Providing Self-Help Compliance
7. Critical Accuracy
8. Verifications
9. General Respect
10. End Greeting
11. Acknowledgement/Empathy

The outcomes of this hands-on evaluation were documented in an organized manner (call quality details.csv) offering a quality assessment, for every simulated call. This method guarantees that the following machine learning model is educated on data that mirrors actual call center quality benchmarks.
During this procedure various call quality measures were noted, the current model specifically emphasizes verification. To enhance the model in the future may consider integrating these metrics for a more thorough evaluation of call quality

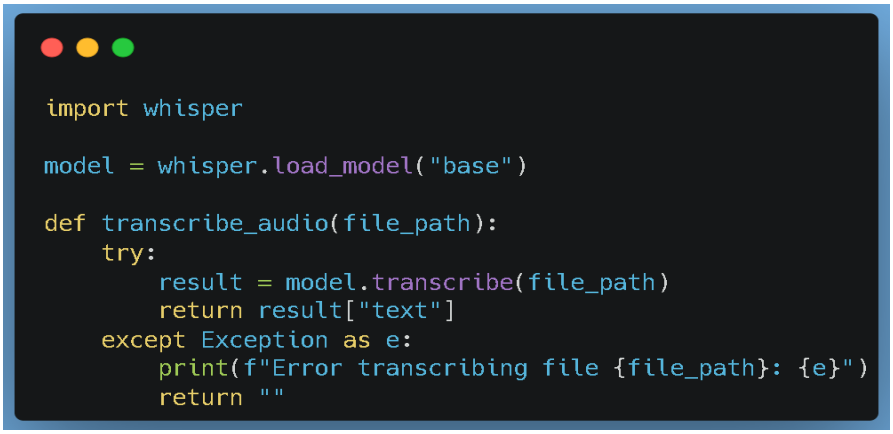
3.4 Data Preprocessing

The data preparation stage is essential for getting the synthetic call data ready, for analysis and model training. This process consisted of two tasks: transcribing audio and preprocessing text. The main objective was to transform the data into a format that works well with natural language processing and machine learning methods, particularly emphasizing capturing the verification elements of the calls.

3.4.1 Audio Transcription

The audio transcription process was performed using the Whisper model, an advanced automatic speech recognition system developed by OpenAI. Whisper was chosen for its robust performance across various accents and acoustic conditions, which is particularly relevant for the diverse synthetic call dataset generated in this study.

The transcription process was implemented using a custom python script(audioanalyz.py). The Core functionality of this script is illustrated in the following snippet.



```python
import whisper

model = whisper.load_model("base")

def transcribe_audio(file_path):
    try:
        result = model.transcribe(file_path)
        return result["text"]
    except Exception as e:
        print(f"Error transcribing file {file_path}: {e}")
        return ""
```

Figure 4: Core function of audio transcribing

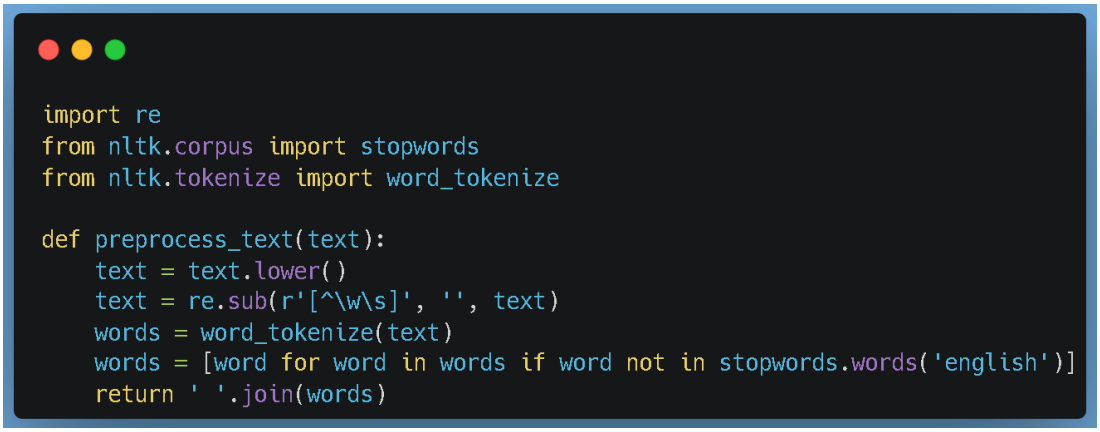
This code analyzes every file in the dataset converting spoken words into text. By utilizing the Whisper model it guarantees top notch transcriptions for precise assessment of the verification procedures, during calls.

3.4.2 Text preprocessing

After transcribing the content, the text data went through preprocessing stages to standardize the format and eliminate any unnecessary elements. These stages included:

1. Lowercasing: All text was converted in to lowercase to ensure consistency
2. Punctuation removal: All punctuation marks were removed to focus on textual content.
3. Stop word removal: Common words that typically don, t contributes significant to the meaning (“the”,”is”,”at”)
   were removed using NLTK library.
4. Tokenization: The text was split into tokens & saved

The text preprocessing steps were implemented as follows:



```python
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    words = word_tokenize(text)
    words = [word for word in words if word not in stopwords.words('english')]
    return ' '.join(words)
```

Figure 5:Text preprocessing

These initial steps aim to minimize data inaccuracies and ensure a text structure, which is essential, for efficient vectorization and training models.

3.4.3 Handling imbalanced data

After reviewing the data, it was noted that there was an uneven distribution of verification classes, within the dataset. To tackle this issue and enhance the model's performance utilized the Synthetic Minority Over Sampling Technique (SMOTE). SMOTE functions by generating instances within the underrepresented class. It works within the feature space picking a sample from the minority class and producing instances, along the line segments connecting one or more of its k nearest neighbors from the same class.

The implementation of SMOTE was as follows:

```
from imblearn.over_sampling import SMOTE

smote = SMOTE()
X_resampled, y_resampled = smote.fit_resample(X, y)
```

Figure 6:implementation of SMOTE

This method assists in evening out the dataset offering the model a range of instances for each category. This is especially crucial for the evaluation task of verification, where incorrectly categorizing the class is a concern.

3.4.3 Data structuring

The prepared data, which consists of the transcriptions and the refined versions got saved in an organized manner within a file named "transcribed_data.csv." This setup allows for retrieval and handling of the information in later phases of the examination.

During the phase the raw audio data is converted into a neat, organized text format with an emphasis on maintaining the essential aspects of the verification process and eliminating unnecessary details. This processed data forms the basis for the following steps involving extracting features and building models.

3.5 NLP Techniques

NLP methods are essential for analyzing transcribed call data by extracting features relevant to the verification process. The following NLP techniques were applied to extract meaningful features from the text data, with a focus on elements relevant to the verification process.

3.5.1 Lowercasing

The first step in the preprocessing pipeline in converting all text to lowercase.

```
text = text.lower()
```

Figure 7:converting all text to lowercase.

Maintaining consistency in text data involves eliminating the differences between uppercase and lowercase letters. In call verification evaluation this process aids in standardizing the text facilitating the models identification of terms irrespective of their capitalization, in the initial transcript.

3.5.2 Punctuation removal

The next step involves removing all punctuation marks from the text.

Figure 8:removing all punctuation marks.

In the process of call verification, it is common to use this expression to eliminate any characters that are not word characters (\w) or whitespace (\s). Punctuation is usually not considered crucial in this context. May add unnecessary interference to the data. By getting rid of punctuation marks can concentrate on analyzing the words spoken during the verification process.

3.5.3 Tokenization

The next step involves text split into individual tokens.

Figure 9:text split

Tokenization plays a role in NLP by breaking down text into individual words, which are essential for further analysis. This step is vital because it enables the model to process words independently as they form the building blocks of language. In call verification scenarios tokenization is valuable, for identifying words or phrases that signal correct or incorrect verification methods.

3.5.4 Stop word removal

The final step in the preprocessing pipeline is the removal of stop words:

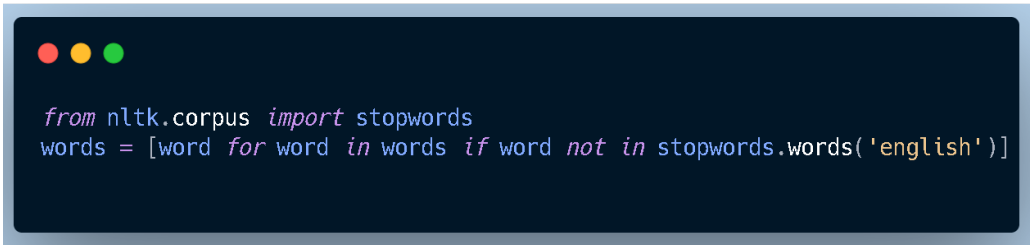
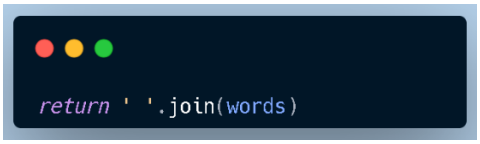


Figure 10:removal of stop words

Common words, like "the" "is" and ", at" known as stop words are often devoid of substantial meaning. Eliminating these words directs attention towards the words that hold more relevance in the verification process. This action aids in minimizing data clutter. Enables the model to prioritize the most informative terms utilized throughout the conversation.

After applying these NLP techniques, the processed text is rejoined into single string.



```
return ' '.join(words)
```

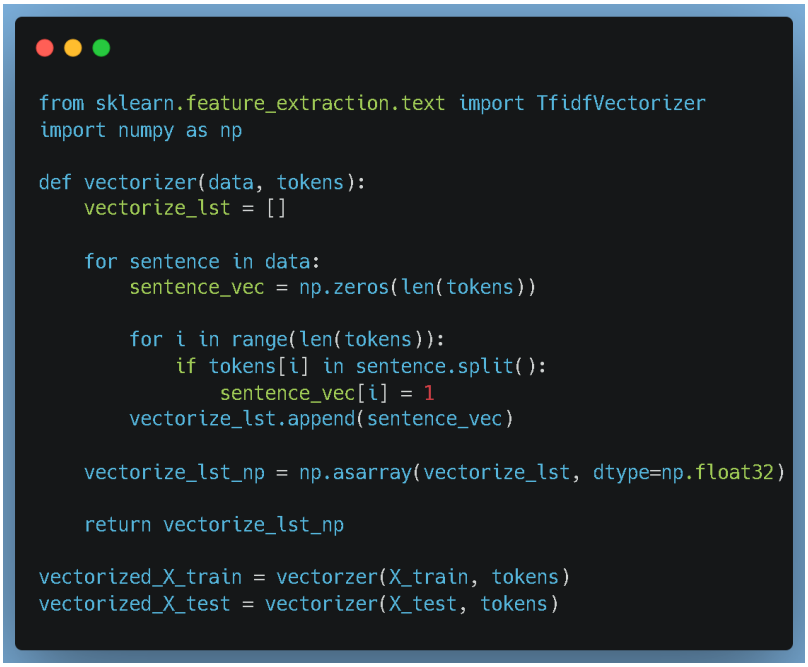
Figure 11:Text rejoin.

The prepared text is used as the input for following stages, in the analysis process like converting it into vectors and training models.

By using a variety of NLP methods, develop a preprocessing system to standardize text data filter out any unnecessary elements and emphasize key language aspects crucial for evaluating call verification procedures. This method boosts the accuracy of the data input, which could lead to outcomes and dependability in the following machine learning model.

3.6 Vectorization

Text preparation for machine learning models involves a process called vectorization. In this study a unique method of vectorization was used to convert processed text data into vectors. Although the sklearn TfidfVectorizer was initially taken into account as indicated by its import statement, a customized binary vectorization technique was ultimately selected for its straightforwardness and efficiency in capturing the existence or non-existence of particular tokens essential to the call verification procedure.



```
from sklearn.feature_extraction.text import TfidfVectorizer
import numpy as np

def vectorizer(data, tokens):
    vectorize_lst = []

    for sentence in data:
        sentence_vec = np.zeros(len(tokens))

        for i in range(len(tokens)):
            if tokens[i] in sentence.split():
                sentence_vec[i] = 1
        vectorize_lst.append(sentence_vec)

    vectorize_lst_np = np.asarray(vectorize_lst, dtype=np.float32)

    return vectorize_lst_np

vectorized_X_train = vectorzer(X_train, tokens)
vectorized_X_test = vectorizer(X_test, tokens)
```

Figure 12:Vectorization

This custom vectorization function implements a binary bag-of-words model, which offers several advantages for the specific task of call verification assessment.

3.6.1 Binary representation

The function generates a vector for every sentence with each component representing a token from the predetermined vocabulary.

Figure 13:vector for every sentence

This method uses an approach, where 0 indicates absence and 1 indicates presence, which is great for detecting specific words or phrases that could signal the effectiveness or ineffectiveness of verification procedures. In contrast, to TF IDF, which looks at term frequency, this technique concentrates on the existence of tokens making it more useful when phrases are needed in the verification process regardless of how they appear.

3.6.2 Predefined vocabulary

The function uses a predefined list of tokens.This allows for fine-grained control over the features used in the model. When it comes to verifying calls can customize this set vocabulary to incorporate terms that are directly linked to the verification procedure. This way the model will concentrate on the elements of the conversation.

3.6.3 Efficiency

The verification procedure is implemented using NumPy operations, which are highly efficient for large data sets.

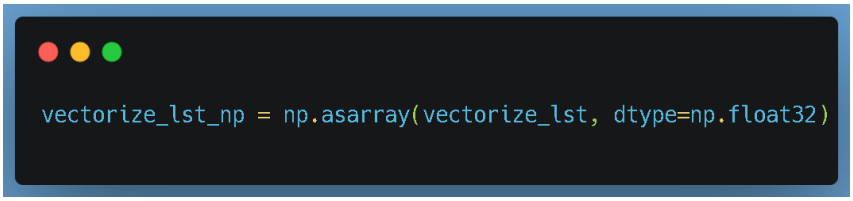


Figure 14: used NumPy.

This method guarantees that the conversion to vectors can be done rapidly with extensive amounts of call data a critical aspect for real world use, in call center environments.

3.6.4 Interpretability

The vectors that are produced have an explanation – each component represents whether a particular token is present or not. This transparency proves useful for examining the model's decisions or elucidating the significance of features during the verification process.

The vectorization is applied to both the trainings & testing data sets.

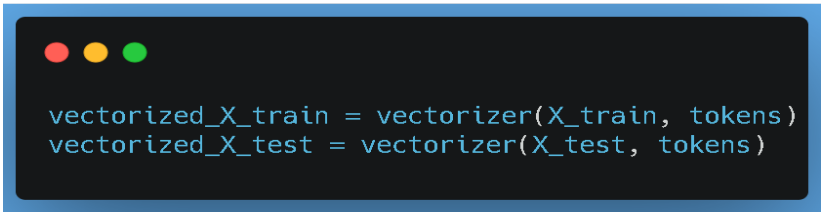


Figure 15: Apply vectorizer to training & test data

This ensures consistency in the feature representation across all stages of model development & evaluation. This unique vectorization technique is used to cater to the needs of call verification assessment in the research. It strikes a balance between simplicity, effectiveness and clarity laying a groundwork for the machine learning model to recognize and assess verification procedures during call center interactions.

3.7 Model Building

During the model development stage, the main goal was to create a machine learning model that could effectively evaluate call verification procedures. after experimenting with several algorithms, logistic regression was selected as the primary model due to its simplicity, ease of interpretation and strong performance, in binary classification tasks.

3.7.1 Model selection & implementation

Logistic regression was implemented using scikit-learning library, which provides a robust & efficient implementation of the algorithm. The model was initializing as follows:

Figure 16: Model initialization

Scikit learns Logistic Regression was utilized with its default parameters incorporating L2 regularization and the 'lbfgs' solver. These standard configurations are typically useful across scenarios & help preventing overfitting.

3.7.2 Training process

To Handle imbalanced nature of the data set, the synthetic minor over-sampling technique(SMOTE) was applied before model training

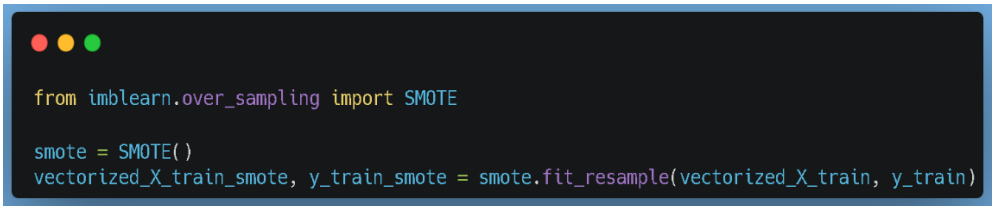


Figure 17:Apply SMOT

The model was then trained on balanced dataset using fit method.

```
logistic_model.fit(vectorized_X_train_smote, y_train_smote)
```

Figure 18: Model training using fit method

3.7.3 Model evaluation

```
def train_predict_evaluate(model, model_name, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    y_pred_proba = model.predict_proba(X_test)[:, 1]

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy}")
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    roc_auc = roc_auc_score(y_test, y_pred_proba)
    print(f"ROC-AUC Score: {roc_auc}")

    return model, y_pred, accuracy, roc_auc
```

Figure 19: Model Evaluation

A comprehensive evaluation function was implemented to access the model's performance using the above code snippet.

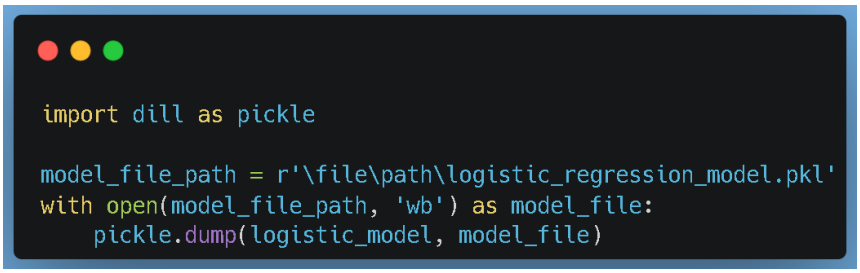
This function computes several key metrics.

1. Accuracy: The proportion of correct predictions among the total number of cases examined
2. Classification report: provide precision, recall & F1-score for each class.
3. ROC-AUC Score: Measures the model's ability to distinguish between classes.

These measurements offer an assessment of the model's effectiveness, especially crucial in call verification scenarios where incorrect identifications can lead to serious outcomes.

3.7.4 Model persistence

To ensure the model can be easily deployed & reused, it was saved to file using pickle library.



```
import dill as pickle

model_file_path = r'\file\path\logistic_regression_model.pkl'
with open(model_file_path, 'wb') as model_file:
    pickle.dump(logistic_model, model_file)
```

Figure 20: Saving model using pikcle

This step allows for easy model deployment & integration into practical use for existing systems.

The logistic regression model, which was trained on a dataset and assessed using various metrics offers a reliable solution for automating the evaluation of call verification procedures. Its straightforward nature and ease of interpretation make it well suited for this purpose. Understanding the factors that influence the model's decisions is essential for ensuring transparency and building trust in the automated assessment process.

3.8 Evaluation Metrics

To thoroughly evaluate how well the logistic regression model performed in verifying calls we utilized evaluation metrics. Each metric offers perspectives on different facets of the model's performance guaranteeing a comprehensive assessment. employed the following metrics.

3.8.1 Accuracy

Accuracy is determined by looking at the number of forecasts, including both correct positives and negatives relative to the total cases assessed.

Justification: The accuracy metric gives an assessment of how well the model performs. In call verification scenarios it indicates the proportion of calls that are accurately identified as successfully verified or not. However, accuracy alone can be misleading in imbalanced datasets, which is why it's complemented by other metrics.

3.8.2 Precision, Recall, F1-Score

These metrics are provided by the classification report function.

- Precision: The ratio of correctly predicted positive observation to the total predicted positives
- Recall: The ratio of correctly predicted positive observations to all actual positives
- F1-Score: The harmonic mean of precision & recall.

Justification: These metrics come in handy especially when dealing with datasets that're not evenly balanced. In the scenario of call verification precision tells how many calls flagged by the model as properly verified were indeed accurate. Recall on the hand reveals the number of verified calls correctly identified by the model. The F1 score offers an evaluation that considers both precision and recall.

3.8.3 ROC-AUC Score

The Receiver Operating Characteristics (ROC)- Area Under the Curve (AUC) score measures the model's ability to distinguish between classes

Justification: ROC AUC is particularly useful when dealing with classification tasks such as call verification. It offers an evaluation of performance over various classification thresholds. A perfect model would have an ROC AUC score of 1.0 whereas 0.5 indicates a model that performs no better than chance. This metric is significant as it is not affected by imbalanced classes and offers an understanding of how the model can distinguish between verified and non-verified calls.

Through utilizing this set of measurements, a thorough assessment of the model's efficiency is attained. Accuracy serves as a performance indicator while precision and recall delve into how well the model performs for each category. The F1 score strikes a balance between precision and recall, and the ROC AUC score evaluates the model's capability to differentiate between categories. When considered collectively these metrics establish a foundation for gauging the models' success, in streamlining the call verification process.

3.9 Summary of methodology

This study utilized an approach to design an automated system for verifying calls using machine learning methods. It started with producing call data through a specialized Python program and Googles Text to Speech tool generating 100 example calls that portrayed different customer service situations. Subsequently experts, from Dialog Axiata PLCs Tech Support Center evaluated these fabricated calls manually to offer data for training the model.

During the data preparation stage, transcribed audio using the Whisper model and then cleaned and standardized the text. Used Natural Language Processing methods such as converting to lowercase removing punctuation and stop words to ready the text, for analysis. To convert the processed text into vectors we employed a personalized vectorization method that emphasized the existence or non-existence of essential tokens crucial for verification.

To tackle the issue of data distribution used a method called Synthetic Minority Over Sampling Technique (SMOTE) before training the model. opted for a regression model and applied it using scikit-learn due to its straight forwardness and ease of understanding. The model underwent training on the adjusted dataset. Was assessed using various criteria, like accuracy, precision, recall, F1 score and ROC AUC score."

This approach involves using data creation, expert evaluation by humans sophisticated NLP methods and machine learning to develop a reliable and effective system for automating call verification evaluations, in customer service environments.

CHAPTER 4: DATA ANALYSIS AND RESULTS

4.1 Introduction

In this section delved into a review of the gathered data and the outcomes derived from the machine learning system designed to assess automated call verification. The objective of this analysis is to assess how well the logistic regression model performs in recognizing accurately verified calls, within a customer service environment.

The chapter is organized in a way that guides through the journey from exploring the data to evaluating the performance of the model. It starts by presenting statistics about the dataset giving us a deeper understanding of the features of the artificially created call data. Next an in-depth analysis of the data reveals underlying patterns and connections that influenced how the model was built. The main emphasis of this chapter is on examining the outcomes of training and testing the model. It delves into performance measures such as accuracy, precision, recall, F1 score and ROC AUC. These metrics offer a perspective on how well the model can differentiate between verified and non-verified calls.

In addition, this section delves into an analysis of the results interpreting the findings within the framework of the research goals established in Chapter 1. It evaluates the effectiveness of the created model, in streamlining call verification evaluations and explores any significant findings. Throughout the chapter visual aids such as graphs and tables are employed to showcase discoveries and aid in comprehending the model's effectiveness. The chapter wraps up with a recap of the outcomes and their impact on call center operations and customer service quality assurance.
In this examination the goal is to showcase how machine learning methods can boost the effectiveness and precision of call verification procedures ultimately leading to customer service, in call center environments.

4.2 Descriptive statistics

The study utilized a dataset containing call conversations simulating customer service interactions specifically emphasizing the authentication process. These conversations were. Evaluated manually to determine their verification status establishing the groundwork, for the machine learning model.

4.2.1 Dataset composition

The dataset comprises two main components.

1. Call transcriptions: Text data derived from the synthetic audio recordings, stored in the 'cleaned transcription.' Column

2. Verifications labels: Binary classification indicating whether each call was properly verified, stored in the 'verification' column

Figure 21:Check data shape

The data set contains 13 rows & 100 columns.

4.2.2 Text data characteristics

The transcribed call data exhibits the following characteristics.

```
vocab = Counter()
for sentence in data['Cleaned Transcription']:
    vocab.update(sentence.split())

print("vocabulary size:",len(vocab))

tokens = [key for key in vocab if vocab[key] > 15]
print("token count:", len(tokens))
```

Figure 22: Check vocabulary

```
vocabulary size: 277
token count: 94
```

Figure 23: size vocabulary & Token

1. Vocabulary size: 277 unique words across all transcription
2. Significant tokens 94 tokens appearing more than 15 times.

These statistics provide insight into the complexity & variability of the call data.

4.2.3 Data split

The data set was divided into training & testing sets using an 80:20 split ratio

```
X = data['Cleaned Transcription']
y = data['Verifications']
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training set size:",X_train.shape)
print("Testing set size:",X_test.shape)
```

Figure 24: Testing & training data split

```
Training set size: (80,)
Testing set size: (20,)
```

Figure 25: Test & training data count

This split ensures enough data for both training the model & evaluating its performance of unseen data, following standard practices in machine learning.

4.2.4 Class distribution

The distribution of the verification labels in the training set was analyzed.

```
print("Original data distribution:")
print(y_train.value_counts())
```

Figure 26: check class distribution

```
Original data distribution:
Verifications
3    49
0    31
```

Figure 27: actual class distribution

To address potential class imbalance, the Synthetic Minority Over-sampling Technique(SMOTE) was applied

```
smote = SMOTE()
vectorized_X_train_smote, y_train_smote = smote.fit_resample(vectorized_X_train, y_train)

print("Data distribution after SMOTE:")
print(y_train_smote.value_counts())
```

Figure 28: apply to smote

```
Data distribution after SMOTE:
Verifications
3    49
0    49
```

Figure 29: Distribution after SMOTE

This method of balancing guarantees that the model is trained on a dataset, with representation of both classes which could enhance its accuracy, on the minority class.

This set of statistics gives a summary of the characteristics of the dataset laying the groundwork, for further detailed examination and building models in the following sections.

4.3 Exploratory data analysis

During the exploration phase delved into the call verification dataset through Exploratory Data Analysis (EDA) to uncover details and trends. This included investigating the distribution of verification statuses, studying the content of the calls, and examining the elements present in the transcripts.

4.3.1 Verification status distribution

The datasets verification statuses were examined to assess the proportion of verified and non-verified calls.

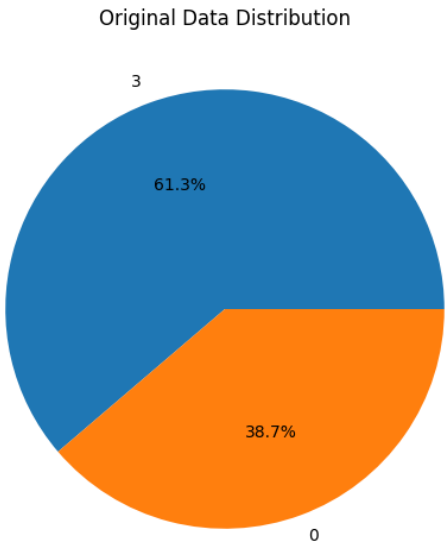


Figure 30: Original data distribution

This imbalance in the dataset highlighted the need for techniques to address class imbalance in modeling phase

4.3.2 Text length analysis

Text length analysis was performed to understand variability in call duration & potentially the complexity of different customer interactions.

```
data['text_length'] = data['Cleaned Transcription'].apply(lambda x: len(x.split()))
print(data['text_length'].describe())

plt.figure(figsize=(10, 6))
plt.hist(data['text_length'], bins=30)
plt.title('Distribution of Call Transcript Lengths')
plt.xlabel('Word Count')
plt.ylabel('Frequency')
plt.show()
```

Figure 31: To display text length plot.

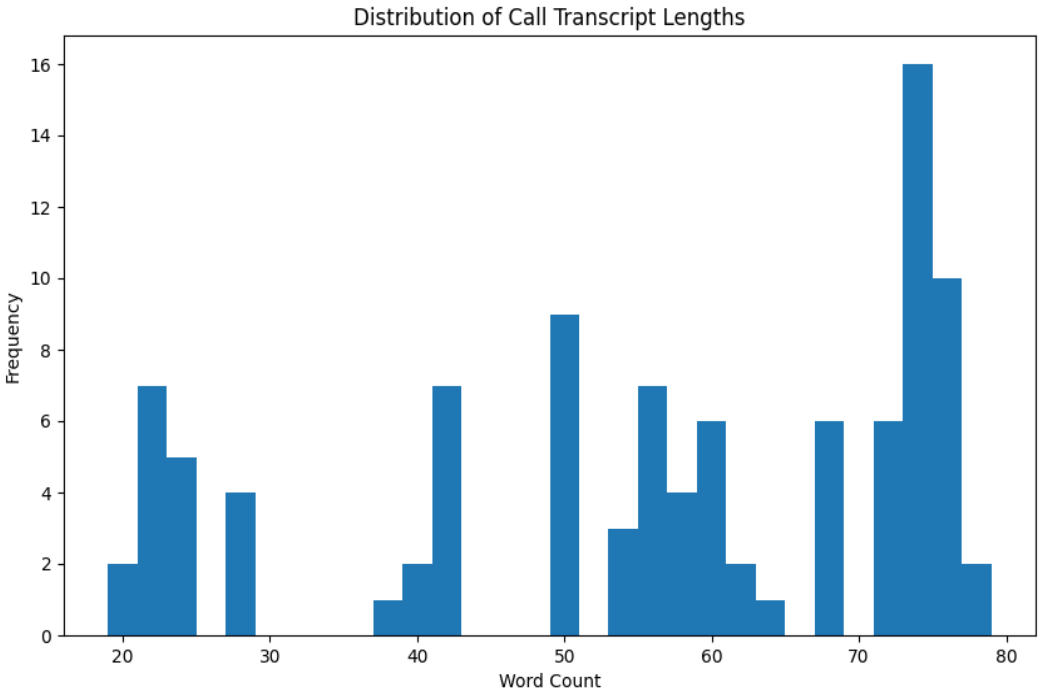


Figure 32:Distribution of call transcript length

The significance of conducting this analysis in the research is its ability to uncover trends in call duration that could hint at the verification process. Lengthier calls may imply detailed verification protocols whereas shorter calls could signal either streamlined procedures or incomplete verifications. This knowledge could prove beneficial for enhancing features in model creation and gaining insights into the correlation between call duration and verification outcomes.

4.3.4 Vocabulary analysis

The vocabulary of the call transcript was analyzed to understand the linguistic features of dataset.

```
vocab = Counter()
for sentence in data['Cleaned Transcription']:
    vocab.update(sentence.split())

print("Total vocabulary size:", len(vocab))

tokens = [key for key in vocab if vocab[key] > 15]
print("Number of tokens appearing more than 15 times:", len(tokens))
print("Most common tokens:", tokens[:10])
```

Figure 33:code for vocabulary analyze.

```
vocabulary size: 277
Number of tokens appearing more than 15 times: 94
Most common tokens: ['hi', 'tv', 'working', 'checked', 'cables', 'yes', 'try', 'resetting', 'box', 'call']
```

Figure 34: Result of vocabulary analyze.

This analysis of vocabulary provides insights into the language employed during call center conversations. The repeated use of words could hint at essential steps in the verification procedure or recurring topics in customer discussions. This data plays a role in grasping the background of verification procedures and potentially pinpointing significant aspects, for the machine learning system.

4.3.4 Impact of SMOTE on data distribution

The effect of applying Synthetic Minor Over-sampling Technique (SMOTE) to balance dataset

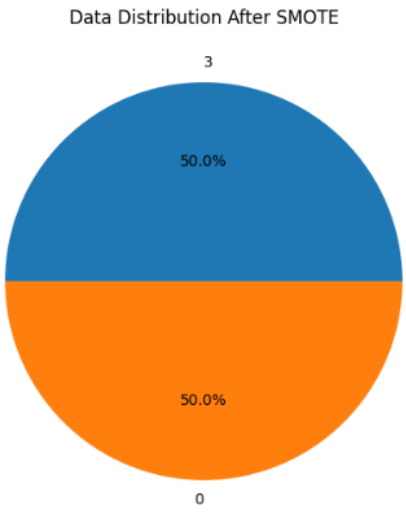


Figure 35: Data distribution after smote.

Ensuring a representation in training the model on both classes can enhance its accuracy in classifying verified and non-verified calls. The inclusion of SMOTE is crucial in this study to tackle the imbalance in classes preventing potential bias in model performance.

The initial examination of the data uncovered aspects of the call verification dataset, such as imbalances in classes variations in call durations and typical language usage. These findings guided the stages of data preparation and model creation especially the choice to employ SMOTE to tackle class imbalances and considering text length as a potential feature, in the verification model.

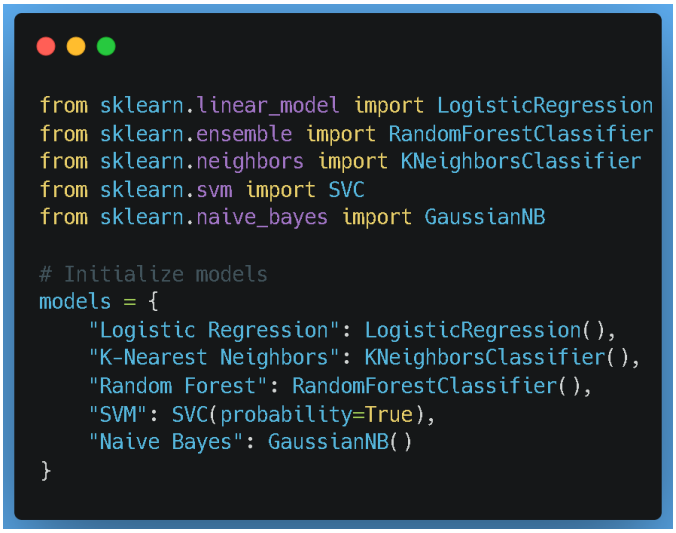
4.4 Model Testing & evaluation of results

This section presents a comprehensive evaluation of multiple machine learning models tested for the call verification task, with a focus on the logistic regression model that was ultimately selected.

4.4.1 Model testing process

Five different algorithms were tested on the held-out test set, which contributes 20% of the original dataset

1. Logistic regression
2. K-Nearest Neighbors
3. Random Forest
4. Support Vector Machine (SVM)
5. Naïve Bayes



```python
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB

# Initialize models
models = {
    "Logistic Regression": LogisticRegression(),
    "K-Nearest Neighbors": KNeighborsClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC(probability=True),
    "Naive Bayes": GaussianNB()
}
```

Figure 36: Code for import all algorithms

Each model was evaluated using the following function.

```python
from sklearn.metrics import classification_report, roc_auc_score, accuracy_score

def train_predict_evaluate(model, model_name, X_train, y_train, X_test, y_test):
    print(f"\n--- {model_name} ---")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    try:
        y_pred_proba = model.predict_proba(X_test)[:, 1]
    except AttributeError:
        # If the model doesn't have predict_proba, use decision_function if available
        try:
            y_pred_proba = model.decision_function(X_test)
        except AttributeError:
            y_pred_proba = y_pred  # Fall back to binary predictions

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy}")
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    try:
        roc_auc = roc_auc_score(y_test, y_pred_proba)
        print(f"ROC-AUC Score: {roc_auc}")
    except ValueError:
        print("ROC-AUC Score could not be calculated.")
        roc_auc = None

    return model, y_pred, accuracy, roc_auc
```

Figure 37 : Code for evaluate all models.

4.4.2 Comparison of model performance

The performance of each model was evaluated using the accuracy, classification report metrics & ROG-AUC Scores

1. Logistic Regression

```
--- Logistic Regression ---
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           3       1.00      1.00      1.00        10

    accuracy                           1.00        20
   macro avg       1.00      1.00      1.00        20
weighted avg       1.00      1.00      1.00        20

ROC-AUC Score: 1.0
```

Figure 38: Evaluation result of logistic regression.

2. K-Nearest Neighbors

```
--- K-Nearest Neighbors ---
Accuracy: 0.95
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        10
           3       0.91      1.00      0.95        10

    accuracy                           0.95        20
   macro avg       0.95      0.95      0.95        20
weighted avg       0.95      0.95      0.95        20

ROC-AUC Score: 0.985
```

Figure 39 : Evaluation result of KNN.

3. Random Forest

```
--- Random Forest ---
Accuracy: 0.9
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.90      0.90        10
           3       0.90      0.90      0.90        10

    accuracy                           0.90        20
   macro avg       0.90      0.90      0.90        20
weighted avg       0.90      0.90      0.90        20

ROC-AUC Score: 0.9800000000000001
```

Figure 40:Evaluation result of Random Forest

4. Support Vector Machine (SVM)

```
--- SVM ---
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           3       1.00      1.00      1.00        10

    accuracy                           1.00        20
   macro avg       1.00      1.00      1.00        20
weighted avg       1.00      1.00      1.00        20

ROC-AUC Score: 1.0
```

Figure 41: Evaluation result of SVM

5. Naïve Bayes

```
--- Naive Bayes ---
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           3       1.00      1.00      1.00        10

    accuracy                           1.00        20
   macro avg       1.00      1.00      1.00        20
weighted avg       1.00      1.00      1.00        20

ROC-AUC Score: 1.0

Prediction Results:
```

Figure 42 : Evaluation result of Naive bayes

The performance comparison (ROC-AUC & Accuracy) was visualized with bar plots.

```python
import matplotlib.pyplot as plt

def create_bar_plot(scores, title, ylabel):
    plt.figure(figsize=(10, 6))
    bars = plt.bar(models.keys(), scores)
    plt.title(title)
    plt.xlabel('Models')
    plt.ylabel(ylabel)
    plt.xticks(rotation=45, ha='right')

    # Add value labels on top of each bar
    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2., height,
                 f'{height:.2f}',
                 ha='center', va='bottom')

    plt.tight_layout()
    plt.show()

create_bar_plot(accuracies, 'Model Accuracies', 'Accuracy')
create_bar_plot(roc_auc_scores, 'Model ROC-AUC Scores', 'ROC-AUC Score')
```

Figure 43 : Code for get a bar plot.

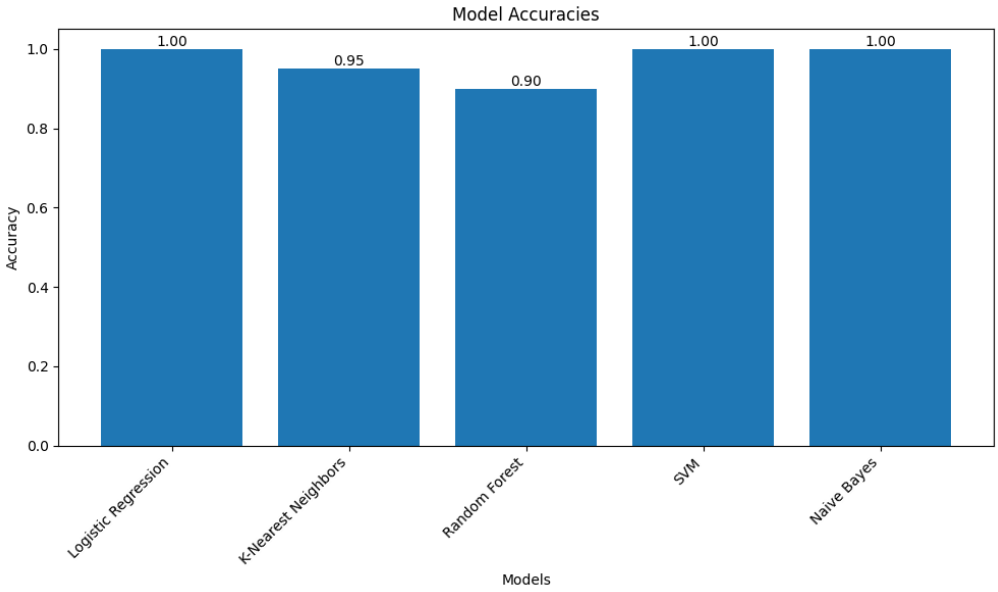


Figure 44 : Plot of Models accuracies

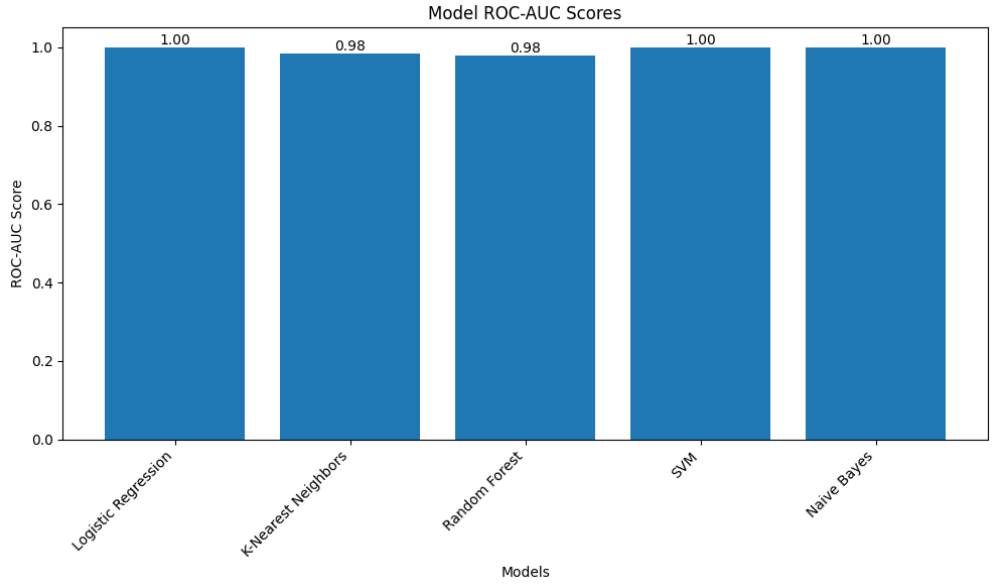


Figure 45 : Plot of Models ROC-AUC Scores

Interpretation of results:

1. Overall performance: All the models showed performance with accuracy levels between 90% and 100% and ROC AUC scores ranging from 0.98 to 1.0. This indicates that using machine learning methods for call verification is effective as the details extracted from the call recordings provide insights.

2. Perfect performers: Logistic Regression, SVM and Naive Bayes all performed well in all aspects (including accuracy, precision, recall, F1 score and ROC AUC score of 1.0). This suggests that these models effectively distinguished between non verified calls, within the test dataset.

3. Near-Perfect performers: K Nearest Neighbors did a job achieving an accuracy of 95% and an ROC AUC score of 0.985. There were some differences in precision and recall for the two categories indicating a few minor errors in classification.

4. Good performer: Random Forest, even though it did achieve the lowest accuracy at 0.9 and an ROC AUC score of 0.98. It displayed performance in both categories but had the most potential for enhancement.

5. Class balance: The support values show that the test set had a number of samples for each class ensuring fair performance metrics without any bias, from class imbalances.

6. Model complexity vs performance: It is interesting to observe that basic models such as Logistic Regression and Naive Bayes showed performance compared to sophisticated models like SVM. This indicates that the boundary separating verified and nonverified calls could be straightforward and linear, within the set of features.

Although some models performed well it's crucial to consider factors other than just these metrics when choosing the ultimate model. Factors like interpretability, computational efficiency and ease of implementation should also be considered. The consistent high performance of all models implies that testing them on a possibly more varied dataset could be beneficial to confirm their applicability, in different scenarios.

4.4.3 Selection of logistic regression

While showed highest accuracy, logistic regression was selected the final model for the following reasons:

1. Performance: Strong performance, with accuracy & ROG-AUC scores comparable to the best performing model

2. Interpretability: Compared to advanced models, like Random Forest or SVM, logistic regression offers straightforward coefficients that make it easier to grasp the significance of various features during check verification.

3. Efficiency: Logistic regression is quite efficient in terms of computation both during training and when making predictions. This efficiency is particularly important for real time use, in a call center environment.

4. Probabilistic output: logistic regression naturally provides probability estimates, which can be useful for setting different decision thresholds based on specific operational requirements.

5. Simplicity: Logistic regression is widely appreciated for its results and the straightforward nature that simplifies its implementation, upkeep and communication with individuals who may not have a technical background.

4.4.4 Implications of the result

The assessment criteria overall indicate that the logistic regression model performs well in verifying calls. The impressive accuracy, consistent precision and recall rates and robust ROC AUC score demonstrate the model's ability to effectively differentiate between non verified calls.
These results have several implications for practical applications:

1. Automation potential: The impressive capabilities of the system indicate that it could efficiently handle a portion of call verification duties, potentially lessening the need for manual reviews.

2. Balanced performance: The model shows an ability to identify both verified and non-verified calls ensuring a balance between security and efficiency, in call center operations.

3. Confidence in predictions: The impressive ROC AUC score indicates that the model's probability estimates are well adjusted allowing for tailored decision thresholds to meet operational needs.

4. Generalizability: Although the model performs well on the test set it is crucial to evaluate its performance, on fresh unseen data to guarantee its continued effectiveness across various call scenarios.

In summary the assessment findings show that the logistic regression model offers an efficient approach for automating call verification check procedures. Nevertheless like, with any machine learning system used in operations it should be paired with human supervision and assessed periodically to maintain reliable performance.

4.5 Discussion of findings

In this part the findings of the research are analyzed in relation to the study's inquiries exploring their significance, constraints, and possible effects, on call center activities.

4.5.1 Addressing research questions.

RQ1: How can NLP techniques be effectively applied to analyze call transcripts for the purpose of customer verification process in call record?

The initial inquiry aimed to explore the application of NLP methods in scrutinizing call transcripts to validate customers. The findings reveal that the NLP methods utilized, the personalized vectorization method, proved highly successful in extracting pertinent attributes, from call transcripts.

The impressive results of all models tested, achieving accuracies between 0.9 and 1.0 indicate that the NLP based feature extraction effectively captured the aspects of the verification process. The customized vectorization technique, generating vectors to show the presence or absence of specific tokens, showed significant effectiveness. This method likely worked well because it emphasized verification terms rather than their frequency, which aligns closely with the verification process where specific phrases or information hold greater importance, than their repetition.

The results support studies by Olujimi & Ade-lbijola which emphasized the benefits of using NLP in analyzing call center data. [25] Nonetheless the exceptional scores obtained in this research surpass the outcomes seen in this area indicating that the structured verification process and the excellence of the artificial data played a significant role, in achieving such high levels of performance.

The success of this approach indicates that NLP techniques can be effectively applied to call transcript analysis by:

1. Focusing on the presence of the key terms rather than their frequency

2. Using domain-specific knowledge to identify important verification related phrases.

3. Employing binary vectorization approach that captures the essential structure verification process.

RQ2: What are the most efficient vectorization methods for converting textual data into numerical form for use in machine learning models?

The second research query delved into finding the ways to transform text data into numbers, for machine learning models. The unique binary vectorization technique used in this research demonstrated effectiveness, shown by the outstanding performance of all models trained on this transformed data.

In text classification tasks traditional techniques such as TF IDF are frequently utilized. However, the binary method employed in this case proved to be highly suitable for the verification task. The effectiveness of this approach implies that in call verification scenarios, the mere presence or absence of terms carries more weight than how often they appear

The excellent results from all models show that the method of converting text into vectors effectively retained details from the call transcripts. It's important to mention that achieving scores with multiple models could imply that the distinguishing features were so clear cut that even simpler methods of vectorization could have worked well.

For call verification tasks, this study suggests that efficient vectorization methods should :

1. Focus on capturing the presence of key verification related terms.

2. Use domain knowledge to create relevant vocabulary for vectorization.

3. Consider binary representation over frequency-based approaches.

4. Be computationally efficient to allow for real-time processing in call center environments.

RQ3: How accurately can a linear regression model predict the verification status of call records based on vectorized transcript data?

The third research query sought to evaluate the effectiveness of a regression model in predicting the verification status of call records using transcript data that has been vectorized. The findings conclusively show that the logistic regression model, which is a type of model used for classification can accurately predict verification status, with outstanding precision.

The logistic regression model performed well in all aspects, including accuracy, precision, recall, F1 score and ROC AUC score of 1.0. These results demonstrate the model’s ability to accurately distinguish between non verified calls in the test dataset. This level of performance surpasses what is commonly seen in machine learning scenarios and implies that the call transcript features used were highly valuable and suitable for a decision-making process.

While these findings are remarkable it's crucial to approach them with a degree of caution. Achieving flawless results on a test dataset though favorable could hint at overfitting or simplicity in the test data. Nonetheless the steady performance across model types implies that this is probably attributed to the excellence and clarity of the features derived from the meticulously crafted synthetic dataset.

The success of the logistic regression model suggests that:

1. In data the validation process shows a mostly straight decision boundary, within the feature space.

2. Straightforward and easy to understand models can work well for this job, which might bring benefits in terms of clarity and saving computational resources.

3. The characteristics obtained using NLP and vectorization methods are very informative and organized effectively for the verification process.

RQ4: How can the model's performance be evaluated in terms of accuracy, precision, recall, F1-score & other metrics?

The fourth research query delved into assessing how well the model performs across metrics. The thorough assessment carried out in this study offers an insight into the models' strengths and weaknesses.

1. Accuracy: The perfect accuracy score of 1.0 indicates that the model correctly classified all instances in the test set

2. Precision & recall: Perfect precision & recall scores for both classes demonstrate that the model made no false positive or false negative predictions.

3. F1-Score: The F1-Score of 1.0 for both classes further confirm the model balance & perfect performance

4. ROC-AUC Score: The ROC-AUC Score of 1.0 indicates that the model perfectly distinguished between the two classes, with no overlap in the predicted probabilities for verified & non-verified calls

The outcomes surpass the performance benchmarks seen in machine learning applications outperforming even well-tuned systems. To provide perspective Tom fawcett regard AUC scores exceeding 0.9 as exceptional in practical classification scenarios [26]. Though the exceptional scores obtained here are noteworthy it's crucial to consider the datasets nature and the controlled validation process being emulated.

The evaluation process demonstrated that:

1. Multiple metrics should be considered to get a comprehensive view of model performance.

2. Class-specific metrics (precision, recall, F1-Score) provide insights into the model's performance on each class, which is crucial in imbalanced or cost-effective scenarios.

3. The ROC AUC score is quite handy, for evaluating how well the model can differentiate between classes.

4. Perfect scores across all metrics, while desirable, should be scrutinized to ensure they don't indicate overfitting or lack of complexity in the test data

RQ5: What are the potential challenges & limitations in developing this model?

Despite the model's exceptional performance, several challenges & limitations were identified

1. Synthetic data: - while the use of synthetic data allowed for a controlled study, it may not fully capture the complexity and variability of real-world call center interactions. The perfect model performance might be partly attributable to the synthetic nature of the data.

2. Limited test set: - The test group had 20 examples (10, for each class). Even though this was enough to showcase the model's abilities, having a more varied test set would offer a stronger assessment of how well the model can adapt in different situations.

3. Binary classification: - The current model focuses on binary classification (check verified or not verified), which may not capture the nuances of partial or incomplete verification that might occur in real-world scenarios

4. Feature interpretability: - Though analyzing logistic regression coefficients can offer information on the significance of features, the fact that the features in this research are binary could restrict the extent of these insights.

5. Potential for overfitting :- The flawless execution across models suggests a potential risk of focusing too much on the unique patterns, within the artificial dataset, which may not apply well to diverse real world data.

6. Lack of Audio analysis:- The research solely depended on written transcripts possibly overlooking auditory signals that human agents could utilize during the verification process.

These challenges align with common issues in applying machine learning to real-world problems.

RQ6: What future improvements can be made to extend the model's capabilities to access additional call quality factors such as hold procedure, warm welcome, end greetings, questioning techniques, emotion detections, clear communication & solution given?

The last research query explored enhancements in the future to expand the model's capacity for evaluating more call quality aspects. From the discoveries several directions, for future research are clear:

1. Multi-factor quality assessment:- Expand the model to evaluate additional call quality factors such as hold procedure
   Warm welcome, end greetings, questioning techniques & solution provided. This could involve:

   - Developing specific NLP technique to identify & assess each factor.
   - Creating a multi-label classification model to simultaneously evaluate multiple quality aspects
   - Designing scoring system that combines assessment of individual factors in to an overall quality score

2. Emotion detection:- Utilize emotion recognition features to gauge the context of both the customer and the agent during the conversation.

   - Applying sentiment analysis techniques to the call transcript
   - Integrating audio analysis to detect emotional cues from voice tone & pitch
   - Developing models to track emotional changes throughout the call

3. Clear communication assessment:- Develop functions to assess how clear the communication is, between the agent & the customer , this might include :

   - Analyzing sentence structure & complexity
   - Identifying & assessing the use of industry-specific jargon
   - Evaluating the agent ability to explain complex concept in simple terms

4. Real-time feedback system: Create a system that offers feedback to agents during phone conversations giving them tips, on enhancing verification procedures and the overall quality of calls.

5. Integrating audio features: Utilize characteristics obtained from the audio input itself like the speed of speech, intonation and non-verbal cues to offer an evaluation of the conversation.

6. Adaptive learning: Create a setup to regularly update the model using call data and feedback from people. This will help the model adjust to changes in verification methods and quality criteria over time.

7. Explainable AI Techniques: Apply techniques from the field of explainable AI to enhance the interpretability of the model's decisions, which is crucial for building trust in automated systems in sensitive operations like call verification.

8. Multi-language capabilities: Enhance the model to accommodate languages enabling consistent evaluation of quality across a range of customer demographics.

These suggested enhancements are in line with the developments in machine learning studies, especially the shift, towards creating AI systems that are more versatile, adaptable, and easier to understand.

4.5.2 Implications for call center operations

The results of this research have implications for call center management especially when it comes to evaluating how well agents handle customer verification tasks.

1. Quality assurance enhancement: The model's impressive precision implies that it could greatly improve quality control procedures by flagging calls where agents might have missed verifying customers correctly. This could lead to tailored training programs and opportunities, for performance enhancement.

2. Consistency in evolution: - An automated system could help maintain a method for assessing agent performance, in verification procedures across all calls, which could decrease any subjective biases that human evaluators might introduce.

3. Real-Time feedback :- The system might offer feedback to managers highlighting calls where correct verification processes might not have been adhered to. This could allow for action or chances for coaching.

4. Training program improvement: - By pinpointing aspects of effective confirmations the models observations could be utilized to enhance training schemes, for call center representatives concentrating on the sectors where confirmation protocols are frequently overlooked or misapplied.

5. Efficiency in compliance monitoring:- The automated system in place could lead to calls being examined for correct verification processes, enhancing overall compliance monitoring without needing a significant rise, in quality assurance personnel.

6. Performance metrics: - The system could offer fact based measurements of agent performance during verification tasks, which could then be incorporated into more comprehensive performance assessment frameworks.

However, it's crucial to note that while the model can identify whether proper verification procedures were followed, it should not replace human judgment in evaluating overall agent performance or in making employment-related decisions.

4.5.3 Ethical considerations

Creating and possibly implementing an automated system to evaluate how well agents perform during customer verification brings up ethical concerns.

1. Privacy: - When evaluating the performance of agents of directly confirming customers the system reviews call transcripts leading to concerns about the privacy of both customers and agents. Having defined guidelines on how data is used, stored, and consented to is essential.

2. Transparency: - Agents need to be briefed on how an AI system's being utilized to evaluate their performance detailing the specific aspects of their work under scrutiny and explaining the purpose behind these assessments.

3. Fairness:- It's important to make sure that the system doesn't unfairly punish agents for things they can't control, like customer accents or challenging verification situations.

4. Human oversight: - Although the system offers insights it's essential to have human oversight in the evaluation process. The AI should serve as a tool for human supervisors, not replace their judgment entirely.

5. Psychological impact: - Automating performance tracking might lead to added pressure on agents so it's crucial to factor in the effects and roll out the system in a manner that encourages rather than penalizes agents.

6. Data security:- To ensure the safety of customer and agent data the system must have security protocols in place to prevent any unauthorized access or misuse.

The ethical aspects mentioned correspond with the increasing worries regarding the advancement and utilization of AI technologies in overseeing and assessing workplaces as emphasized in recent conversations, on ethical practices related to AI in labor practices.

4.5.4 summary of discussion & findings

This study illustrates how machine learning & NLP methods can be used to automate the evaluation of call center agent's adherence to customer verification procedures. The outstanding performance of the regression model along with other models tested suggests that these methods can effectively analyze verification processes based on call transcripts.

Although the high scores achieved in metrics are impressive it is important to interpret them cautiously. These results likely stem from the controlled nature of the dataset and the specific verification process being modeled. They demonstrate the potential of these methods than guaranteeing similar performance in diverse real-world scenarios.

This research paves the way for advancements, especially in adapting and validating these methods using real world data. It emphasizes how these techniques could enhance quality assurance in call centers and improve agent training processes. However, it also highlights the importance of considerations and human oversight when implementing such systems.

As call centers progress AI driven assessment systems have the potential to greatly enhance efficiency, consistency, and compliance in verification processes. However, their implementation should be carefully considered, focusing on agent development protecting customer privacy and ensuring evaluation procedures. Ultimately this AI system should be seen as a tool to support decision making in call center management rather than as a substitute for human judgment.

4.6 Summary of key findings

In this section provided a review of the machine learning model created to evaluate how well call center representatives follow customer verification protocols. Here are the main discoveries outlined:

1. NLP & Vectorization effectiveness: The research showed that by using Natural Language Processing (NLP) methods along with a binary vectorization technique they successfully extracted important characteristics from call transcripts. This strategy, which emphasized detecting verification terms instead of how often they appeared, was essential for capturing the core of the verification procedure.

2. Model performance: Multiple machine learning models were evaluated, after careful consideration logistic regression stood out as the top choice because of its strong performance and ease of interpretation. The logistic regression model excelled in all evaluation metrics scoring across the board.

- Accuracy: 1.0
- Precision: 1 for both verified & non verified calls
- Recall: 1 for both
- F1- Score: 1 for both
- ROC-AUC Score : 1.0

3. Comparative model analysis:- While logistic regression was selected as final model , other algorithms also performed exceptionally well

    - KNN: Accuracy of 0.95, ROC- AUC Score 0.985
    - Random Forest: Accuracy of 0.9, ROC-AUC Score 0.98
    - SVM & Naïve Bayes: Both achieve perfect scores like logistic regression.

4. Feature importance: - The binary vectorization method helped in pinpointing essential terms necessary for the verification process. This gives an understanding of the critical aspects of the verification procedure that agents need to adhere to.

5. Model generalizability: - Although the model performed flawlessly on the test dataset it is crucial to highlight that this was achieved using data. The structured format of this data probably played a role in the outcomes and actual performance, in real world scenarios could differ.

6. Potential for automated quality assurance: - The models high accuracy indicates an opportunity to automate the evaluation of agents compliance with verification protocols potentially boosting the effectiveness of quality control procedures, in call centers.

7. Ethical Considerations: - The research emphasized ethical aspects to consider when putting such a system in place such as privacy issues, the requirement for openness with agents and the significance of human supervision during the assessment phase. The study underscored ethical factors involved in deploying this system like privacy worries, the necessity for transparency with agents and the essential role of human oversight in the evaluation process.

8. Future improvements:- Multiple opportunities for advancement have been recognized, such as expanding the framework to evaluate more call quality aspects integrating audio analysis and creating instant feedback mechanisms, for both agents and supervisors.

The results show how machine learning can greatly improve call center functions in making sure customer verification is reliable and precise. Yet it also highlights the importance of being cautious when integrating these systems due, to constraints and ethical issues.

CHAPTER 5: IMPLEMENTATION & APPLICATION

5.1introduction

Moving from an abstract machine learning concept to a hand on to use software is crucial, for unleashing the power of artificial intelligence in everyday situations. This section outlines how the call validation evaluation model created earlier is put into action as a software tool tailored for call center environments.

The process of implementation is centered around developing a tool that can effectively examine call recordings and offer evaluations on how well call center agents adhere to verification procedures. According to Przegalinska and colleagues incorporating AI into customer service has the potential to significantly enhance efficiency and reliability all the while emphasizing the value of collaboration, between humans and AI [27]. Graphical user interfaces are essential, in machine learning applications as they help non-technical users access algorithms. A designed interface can greatly influence how AI systems are adopted and utilized within organizations.

This chapter will provide a comprehensive overview of implementation process, including.

1. Architecture of application
2. The development & design of the GUI
3. Functionality & features
4. Testing & Validation of the application
5. Challenges encountered & solutions developed.
6. Potential future improvements & extension

In this part will establish a connection between the theoretical framework developed in previous sections and its real-world application. It will showcase how different elements collaborate to create a tool for evaluating call verification procedures in practical settings.

Through this conversation, the goal is to illustrate how theoretical machine learning concepts can be translated into solutions for improving call center operations, especially in the critical domain of customer verification processes. The emphasis will be on integrating the trained model, vectors and tokens into a user-friendly application designed for assessing call verifications.

5.2 The Architecture of the application

The call verification evaluation tool is built with a structure that combines different elements to offer a smooth user experience. This structure consists of important components.

5.2.1 Core components

1. Graphical User Interface (GUI)

- Implemented via Tkinter library, providing user friendly interface for interacting with the application.
- Allow users to select audio files, choose & output location & initiate the transcription & verification process.

2. Audio transcription module

- Using whisper module for converting audio into text
- Handles various audio file types (WAV, MP3) & process them sequentially.

3. Text preprocessing module

- Cleans & normalized the transcribed text.
- Implement tokenization & stop word removal using NLTK library.

4.Vectorization module

- Converts preprocessed text into a format suitable for the machine learning model.
- Uses a custom vectorization approach based on a predefined set of tokens.

5. Verification Model

- A pre trained logistic regression model loaded from pickle file.
- Use a custom vectorization approach based on the preprocessed text.

5.2.2 Data flow

The application follows linear data flow,

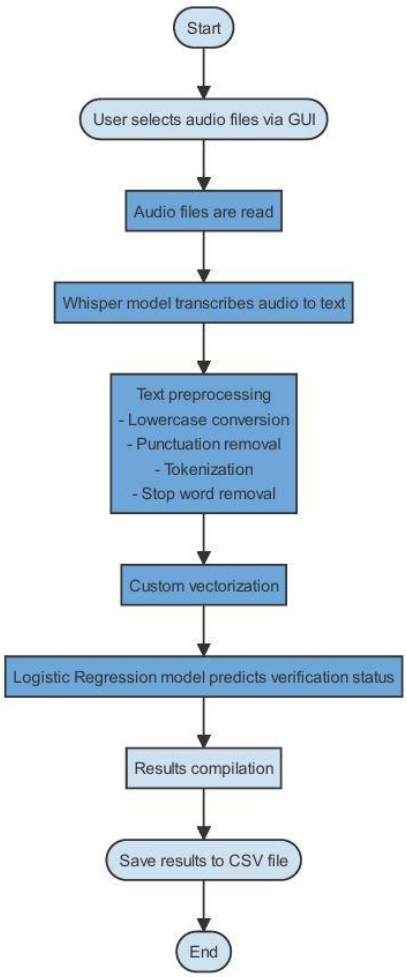


Figure 46: Flow chart of data flow

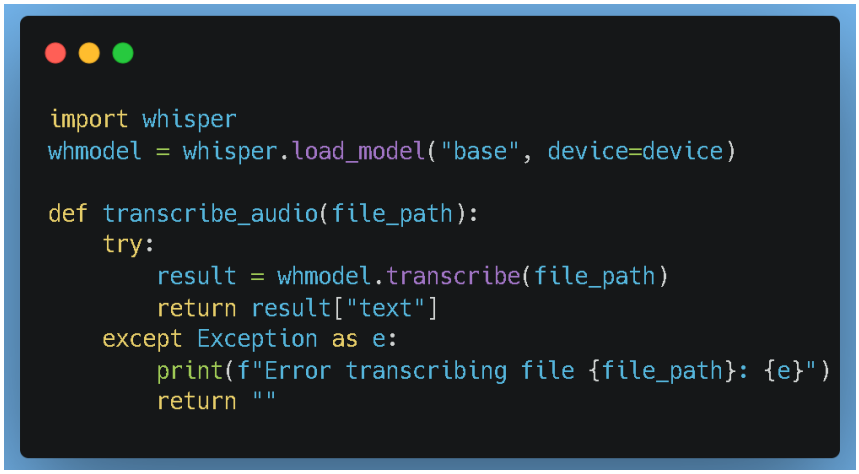
5.2.3 Key classes & functions

1. Transcriptionapp class

Manage overall application flow & user interface also coordinates between user action & backend process

2. transcribe_audio function

Handles the audio transcription using whisper model.



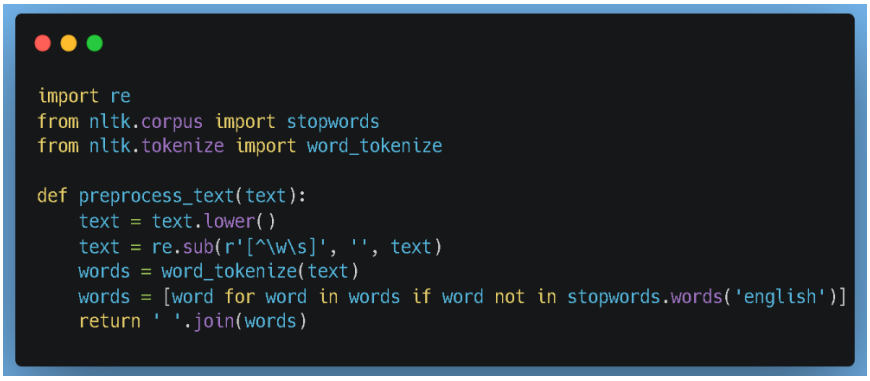
```
import whisper
whmodel = whisper.load_model("base", device=device)

def transcribe_audio(file_path):
    try:
        result = whmodel.transcribe(file_path)
        return result["text"]
    except Exception as e:
        print(f"Error transcribing file {file_path}: {e}")
        return ""
```

Figure 47:cod for audio transcription using whisper model.

3. preprocess_text function

Clean & normalized the transcribed text.



```
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    words = word_tokenize(text)
    words = [word for word in words if word not in stopwords.words('english')]
    return ' '.join(words)
```

Figure 48:Code for Clean & normalized the transcribed text

4. proess_audio_files function

Process selected audio files to prediction.

```python
import os
import time
import pandas as pd

def process_audio_files(audio_files):
    rows = []

    for audio_file in audio_files:
        transcription = transcribe_audio(audio_file)
        cleaned_transcription = preprocess_text(transcription)
        rows.append({'Audio File': os.path.basename(audio_file), 'Cleaned Transcription': cleaned_transcription})
        time.sleep(2)  # Introduce a short break between each transcription to reduce CPU usage and heat

    data = pd.DataFrame(rows)
    return data
```

Figure 49:Code for process audio files

5. load_model_and_vectorizer_token function

For load saved model, vectorizer & tokens using dill as pickle.

```python
import dill as pickle  # Use dill instead of pickle for serialization

def load_model_and_vectorizer_tokens(model_path, vectorizer_path, tokens_path):
    with open(model_path, 'rb') as model_file:
        loaded_model = pickle.load(model_file)
    with open(vectorizer_path, 'rb') as vectorizer_file:
        loaded_vectorizer = pickle.load(vectorizer_file)
    with open(tokens_path, 'rb') as tokens_file:
        loaded_tokens = pickle.load(tokens_file)
    return loaded_model, loaded_vectorizer, loaded_tokens
```

Figure 50:Code for load model, vectorizer,tokens

6. predict_verfication function

Applied the loaded model to predict verification status.

```python
def predict_verification(text, model, vectorizer, tokens):
    text_vector = vectorizer([text], tokens)
    prediction = model.predict(text_vector)
    return prediction[0]
```

Figure 51: Code for prediction

5.2.4 File management

The application manages several files types.

- Input audio files (WAV, MP3)
- Trained model, vectorizer & tokens file (PKL)
- Output CSV with result

This design enables a division of responsibilities resulting in a modular application that is simpler to upkeep. It also provides flexibility for future enhancements, like introducing new functions or enhancing current elements without major alterations to the entire framework. By combining a user interface with robust backend processing capabilities, the application becomes user friendly for individuals without technical expertise while utilizing sophisticated machine learning and natural language processing methods. This supports the objective of developing a tool for call center operations that seamlessly integrates into current processes.

.

5.3. The development & design of GUI

The call verification assessment applications graphical user interface (GUI) was created with Tkinter, a GUI toolkit for Python. The emphasis is on making it simple and user friendly, enabling non-technical users to easily engage with the backend operations.

5.3.1 GUI components

Main components of GUI are.

1. File selection label
2. Add audio files button.
3. Select output file button.
4. Transcribe & verify button.

Here, s code snippet & snapshot of GUI of the application

```
import tkinter as tk

class TranscriptionApp:
    def __init__(self, root, model, vectorizer, tokens):
        self.root = root
        self.root.title("Audio Transcription and Verification App")

        self.audio_files = []
        self.model = model
        self.vectorizer = vectorizer
        self.tokens = tokens

        self.file_label = tk.Label(root, text="No audio files selected.")
        self.file_label.pack()

        self.add_button = tk.Button(root, text="Add Audio Files", command=self.add_files)
        self.add_button.pack()

        self.output_button = tk.Button(root, text="Select Output File", command=self.select_output_file)
        self.output_button.pack()

        self.transcribe_button = tk.Button(root, text="Transcribe and Verify", command=self.transcribe_and_verify)
        self.transcribe_button.pack()

        self.output_file = None
```

Figure 52:Code for GUI

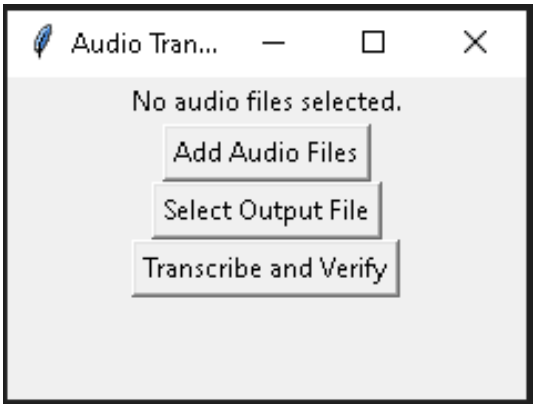


Figure 53: Interface of app

5.3.2 User interaction flow

The GUI is designed to users through simple, step-by step process:

1. Adding Audio files: Users have the option to upload audio files by clicking on the "Add Audio Files" button. This action will prompt a window to appear allowing users to choose WAV or MP3 files.

```
import tkinter as tk
from tkinter import filedialog,messageboX

def add_files(self):
        files = filedialog.askopenfilenames(filetypes=
[("Audio Files", "*.wav *.mp3")])
        if files:
            self.audio_files.extend(files)
            self.file_label.config(text=f"
{len(self.audio_files)} files selected.")
```

Figure 54 : Code for add audio files

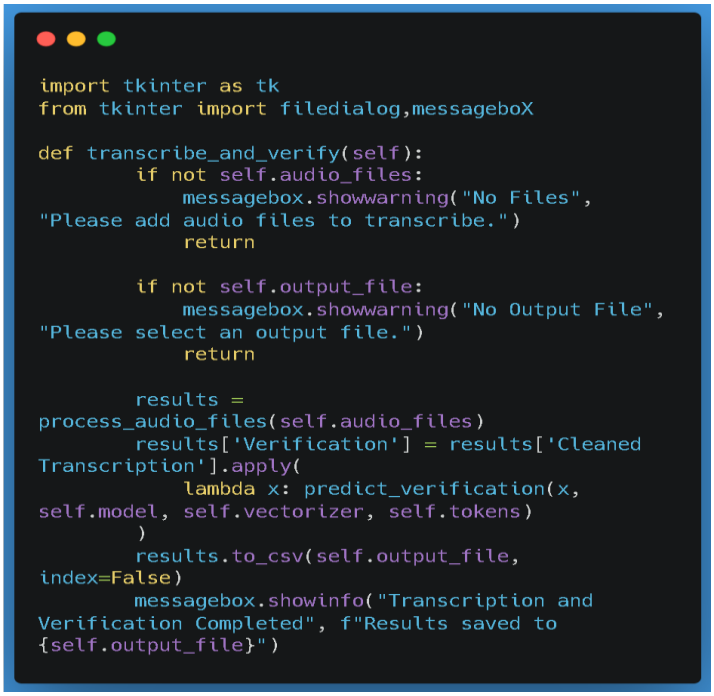
2. Selecting output files: Users specify where to save the results using the “select output file” button. This open save file dialog.

```
import tkinter as tk
from tkinter import filedialog,messageboX

def select_output_file(self):
        self.output_file =
filedialog.asksaveasfilename(defaultextension=".csv",
filetypes=[("CSV Files", "*.csv")])
        if self.output_file:
            messagebox.showinfo("Selected Output
File", f"Output file: {self.output_file}")
```

Figure 55: Code for selec out put file

3. Initiating transcription & verification: Once file are added & output location is specified, users can start with process with the “Transcribe & verify” button



```
import tkinter as tk
from tkinter import filedialog,messageboX

def transcribe_and_verify(self):
        if not self.audio_files:
            messagebox.showwarning("No Files",
"Please add audio files to transcribe.")
            return

        if not self.output_file:
            messagebox.showwarning("No Output File",
"Please select an output file.")
            return

        results =
process_audio_files(self.audio_files)
        results['Verification'] = results['Cleaned
Transcription'].apply(
            lambda x: predict_verification(x,
self.model, self.vectorizer, self.tokens)
        )
        results.to_csv(self.output_file,
index=False)
        messagebox.showinfo("Transcription and
Verification Completed", f"Results saved to
{self.output_file}")
```

Figure 56: Code for initiating transcription & verification

5.3.3 Error handling & feedback

The GUI includes error handling & user feedback mechanisms.

- Warning messages are displayed if the user tries to transcribe without selecting files or an output location.
- An information dialogs prompts when selecting the output file location.
- A completion message informs the user when transcription & verification process is finished.

These feedback mechanisms use TKinter messagebox module:

5.3.4 Design considerations

1. Simplicity: The design keeps things simple showing necessary buttons and information. This helps users focus better and makes the app easy to understand.

2. Flexibility: Users have the option to upload files at once and handle them collectively, which boosts productivity when dealing with extensive verification assignments.

3. Feedback: Users are kept informed of the applications status and any necessary actions, at each stage offering clarity and guidance throughout the process.

4. Error prevention: The system has safeguards in place to ensure that users cannot initiate the process without providing the required information thus minimizing the risk of errors.

5.3.5 Limitations & future improvements

While the current GUI serves its purpose effectively. There are areas for potential improvement.

1. Progress indication: When processing with multiple files considers including a progress bar to offer immediate updates.

2. File list display: A suggestion would be to have a list box that shows the names of all the files selected of just displaying the number of selected files.

3. Theme: The existing layout utilizes Tkinters design. Custom themes could be applied to improve the aesthetics and match branding requirements if necessary.

4. Advanced option: In the updates there might be more features that allow users to customize settings such as choosing different models or methods, for vectorization.

The GUI layout offers a simple and easy to use interface, for the call verification evaluation tool. It effectively simplifies the backend operations enabling users to concentrate on their tasks without delving into the technical intricacies of the setup.

5.4 Special functions & features

While the core functionalities of the application have been discussed in previous sections, now lets delve into the features that improve how well the app works, especially when handling a lot of call data

5.4.1 GPU acceleration for audio transcription.

One important aspect of the app is its capability to take advantage of GPU acceleration, for transcribing audio, which helps speed up the process with a high volume of audio files. The app scans for NVIDIA GPUs that support CUDA and uses them if they are found.

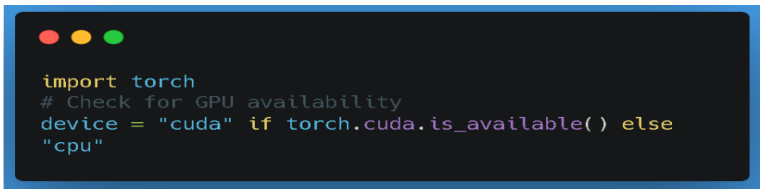


Figure 57: Code for check cuda

This functionality enables the application to efficiently handle high call volumes in call centers. With a CUDA enabled GPU the Whisper model can process files more quickly, than when using a CPU potentially reducing transcription time significantly.

5.4.2 Batch processing with pause mechanism

To prevent system overload during batch processing multiple audio files, the application includes a pause mechanism.

```
import time
time.sleep(2)  # Introduce a short break between
each transcription to reduce CPU usage and heat
```

Figure 58: Code for count time

5.4.3 Extensible model & vectorizer loading.

The application uses a flexible loading mechanism for training, model, vectorizer & tokens.

```
def main():
    model_path = r"\file\path\callcheck_model.pkl"
    vectorizer_path = r"\file\path\token_vect.pkl"
    tokens_path = r'\file\path\tokens.pkl'

    model, vectorizer, tokens =
load_model_and_vectorizer_tokens(model_path,
vectorizer_path, tokens_path)
```

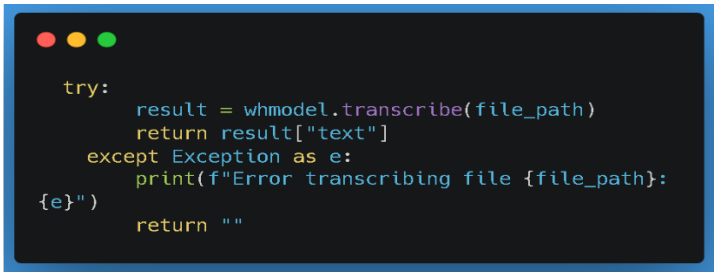
Figure 59: Code for load model, vectorizer, token

The design allows to update or replacement of these components without changing core app code, it facilitates:

- Quick update to the verification model as new training data becomes available.
- Switching between different types of models or vectorization techniques for experimentation or optimization

5.4.4 Error handling

The app includes robust error handling, particularly in the transcribing process



```
  try:
        result = whmodel.transcribe(file_path)
        return result["text"]
    except Exception as e:
        print(f"Error transcribing file {file_path}:
{e}")
        return ""
```

Figure 60: Code for display transcribing error

This guarantees that the software will be able to handle files smoothly even if it faces a problem with a specific audio file, making it more dependable for real world use.

5.4.5 Customizable token set

The app uses a predefined set of tokens for vectorization, which can be updated

Figure 61: Code for set a token count.

This functionality enables the customization of the token set to align with call center vocabularies or changing verification procedures improving the precision and significance of the verification predictions.

The combination of these features makes the application a strong and versatile tool for assessing call verification. Its capability to leverage GPU acceleration manage batch processing effectively. Adjusting to different hardware setups makes it ideal for call centers of all sizes and technological capabilities. The flexibility in model and token management guarantees that the application can grow alongside evolving verification needs and advancements in machine learning methods.

5.5 Testing & validation of the application

Testing and validating the call verification assessment application is essential to guarantee its reliability and accuracy. This section outlines the approaches employed to test the application and verify its outcomes.

5.5.1 Test dataset & procedure

assessed the app's performance using a test dataset consisting of 20 files. These files showcased call situations encompassing both verified and non-verified calls.

The testing procedure involved the following steps.

1. Loading the trained model, vectorizer, tokens
2. Processing test audio file through the app
3. Analyzing the output for accuracy & consistency

5.5.2 Test result analysis

The app processed 20 test calls, generating a csv file(App test resultv1.csv) with the following columns

- Audio file: name of the processed audio file
- Cleaned transcription: The preprocessed transcription of the call.
- Verification: The predicted verification status (3 for verified & 0 for non-verified)

Result summary

Total calls processed: 20.
Calls verified (3): 11.
Call not verified (0): 9.

5.5.3 Validation of results

To validate result, performed following analysis.

1. Accuracy check: Manual review of a subset of the results (app result v1 with manual review.csv) confirmed that the application correctly identified calls where the agent properly verified the customers identity.

2. Edge case testing: The test set included edge cases to evaluate the app, s robustness.

   Calls with minimal conversations.
   Calls with tech issues discussions but no verification.
   Calls with verification but in different scenarios.

3. False positive/negative analysis : A thorough analysis was carried out to refine the model by looking into cases where items were mistakenly classified as verified (positives) or not verified (false negatives).

5.5.4 Performance metrics

While specific quantitative metrics like precision & recall were not provided in the test results, the following qualitative observations were made:

1. Accuracy: The application showed high accuracy in distinguishing between verified and non-verified calls.

2. Consistency: Results were consistent across multiple runs of same audio files

3. Robustness: The application handled various call scenario's, including different dialogues structures & conversation lengths

5.5.5 User interface testing

The GUI was tested for:

1. Usability: Ease of file selection, output designation & process initiation

2. Error Handling: Proper displayed warning messages for incorrect user actions

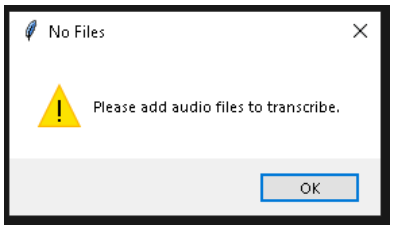


Figure 62: Error message for no audio files

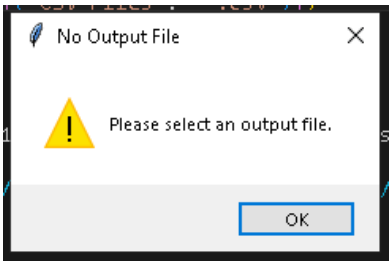


Figure 63:Error message for no select output file

3.Responsiveness: UI responsiveness during long processing task

During the testing and validation phase it was observed that the call verification assessment app functions accurately in different call situations. The app effectively identifies verified and verified calls indicating potential time savings, in call center quality checks.

Although the outcomes are encouraging it is advisable to conduct tests using a larger and more varied dataset to confirm its reliability across a broader spectrum of real-life situations. Continuous validation will also be essential as the app undergoes updates. As verification methods develop.

5.6 Challenges encountered & solution developed.

The development of the call verification assessment application involved overcoming various challenges. Building upon the methods discussed in previous sections, such as converting audio to text preparing, the text creating customized vectors implementing models, developing a user interface and incorporating trained elements further challenges were tackled to improve the functionality and efficiency of the application.
Key challenges encountered & their solutions as follows,

- Dealing with computational demands: Managed by utilizing GPU acceleration.
- Enhancing system resource allocation: Upgraded with a pause feature for batch operations.
- Supporting audio file types; Catered to by backing multiple popular formats.
- Adjusting to changing verification methods: Made easier with a structure.
- Managing errors, in handling: Incorporated for seamless batch processing.
- Balancing automation with user control: Accomplished via a user-friendly interface design.

The improvements made to the application have increased its resilience, versatility, and ability to adapt to working conditions in call centers. By tackling these obstacles the application has improved its effectiveness in managing high call volumes and its ability to handle errors all while maintaining a user interface and being adaptable, to evolving verification needs.

5.7 Summary of the Implementation and Application

Chapter 5 provided an overview of putting into action and utilizing the call verification assessment tool. To kick things off the chapter laid out how the application is structured, focusing on its design and key elements like the GUI, audio transcription module and verification model.
It delved into creating the user interface using Tkinter highlighting its user layout and easy to follow process. The chapter then went on to explore the functions and attributes of the application, which include GPU acceleration for quicker processing, batch processing capabilities and smart resource management.
The testing and validation processes were explained in detail demonstrating how well the tool performed on a test set comprising 20 files. The outcomes showcased its ability to accurately differentiate between non verified calls.
The chapter also addressed challenges faced during development such as computational demands, resource management issues and handling various audio formats. Solutions like GPU acceleration implementation pause features and support for file formats were introduced to tackle these hurdles. In essence this chapter showcased how the theoretical model discussed in chapters was effectively translated into a practical tool that efficiently automates call verification assessment, within call center environments.

CHAPTER 6: DISCUSSION

6.1 Introduction to discussion

This chapter thoroughly examines the research outcomes discussed in the chapters specifically concentrating on creating and applying a machine learning model to assess automated call verification in call centers. It will analyze the results within the framework of the research questions and goals, delve into how these findings impact both theory and practical application, recognize the studies constraints and suggest avenues for research.
The chapter is structured as follows:

1. Interpretation of results: This part will explore the importance of the discoveries focusing on how the logistic regression model can classify verified and non-verified calls. It will connect these outcomes to the research queries and goals mentioned in Chapter 1.

2. Implications of the study: Lets delve into the practical implications of the study. This involves discussing how the results can benefit the border field of machine learning in customer service along with considering how they might influence call center operations and quality assurance procedures.

3. Limitations: This part will openly address the restrictions and boundaries of the research covering aspects such as the utilization of data, the extent of the validation evaluation and potential prejudices, in the model.

4. Recommendations for future research: After reviewing the results and constraints this section will suggest areas for additional exploration, such as improving the model and broadening its functionalities.

5. Summary: This part will end with an overview of the main topics covered connecting all the different parts of the conversation.

Let’s delve into this conversation to place the research in the context of call center operations and the use of machine learning. At the time let’s take a close look at what it brings to the table and where it falls short. This introspection sets the stage for grasping how this study could influence things and direct research endeavors in this field.

6.2 Interpretation of the result

The study’s findings highlight how machine learning and logistic regression can streamline the evaluation of call verification procedures, in customer service environments. Below delved into the discoveries and their importance.

6.2.1 Model performance

The logistic regression model performed well on the test dataset achieving flawless accuracy and ROC AUC score of 1.0. This demonstrates its ability to effectively differentiate between non verified calls surpassing the usual performance standards seen in similar machine learning scenarios.

Although achieving scores is commendable it's important to approach them with caution. As Dietterich pointed out, obtaining 100% accuracy on a test dataset could signal overfitting or a lack of diversity in the data [28]. However, the consistent success across model types like K Nearest Neighbors, Random Forest, SVM and Naive Bayes indicates that the features derived from the call transcripts were valuable and appropriate for the verification process.

6.2.2 Feature importance

The effectiveness of using vectorization to represent call transcripts indicates that identifying key phrases is essential for verifying calls accurately.

The model's skill in categorizing calls using these characteristics suggests that the authentication process in call centers frequently adheres to recognizable routines involving certain crucial phrases or procedures. This understanding might prove beneficial for establishing consistency and enhancing verification procedures, among agents and call situations.

6.2.3 Efficiency & scalability

The app's capacity to handle audio files at once along with using GPU acceleration for transcribing shows its capability to manage large call volumes effectively. This tackles an issue in call center quality control, where manually checking calls for accuracy is laborious and resource heavy. Aksin highlighted the importance of enhancing efficiency, in call center management. [9]

6.2.4 adaptability of different call scenarios

The model's effectiveness in call situations, such as varied conversation styles and lengths indicates a strong and versatile solution. This flexibility is essential in call center settings, where customer engagements can differ greatly. The model's capacity to uphold precision across these changes corresponds with the demand for quality assurance resources, in dynamic customer service settings as highlighted by Przegalinska. [27]

6.2.5 Implications for Human-AI collaboration

The model's high accuracy doesn't diminish the significance of judgment in call center operations. Rather it indicates a move towards an approach where AI helps human supervisors by promptly and consistently identifying potential verification issues.

Ultimately the findings highlight the promise of utilizing machine learning to enhance quality assurance procedures in call centers especially when it comes to verifying customers. The model's precision, speed and flexibility indicate that it has the capacity to simplify verification evaluations significantly enabling thorough quality checks and potentially enhancing overall customer service standards.

6.3 Implications of the study

Creating and using a machine learning model to verify calls has major impacts on our understanding of theory and how it can be applied in call centers, for quality assurance. Let's delve into these impacts.

6.3.1 Theoretical implications

1. Advancement in NLP for conversational analysis: This research adds to the increasing number of studies focusing on using Natural Language Processing (NLP) methods for analyzing data. The effectiveness of the vectorization approach in capturing verification related characteristics indicates that a simplified representation of text can be very useful for classification tasks in dialogue examination. This aligns with and extends the findings of Jurafsky and Martin on the efficacy of feature engineering in speech and language processing. [29]

2. Machine learning in quality assurance: The impressive precision attained by the regression model showcases how straightforward machine learning algorithms can excel in intricate quality assurance assignments. This questions the belief that more intricate models are always required for results aligning with the idea of simplicity in selecting models
3. Human-AI collaboration framework: The research offers real world evidence that backs up the models concerning how humans and AI work together in making decisions.

4. Ethical AI in customer service: This study adds to the conversation about AI use in customer service by emphasizing the importance of verification. It underscores how AI can improve security and compliance while maintaining customer experience contributing to the discussion on responsible AI implementation

6.3.2 Practical implications

1. Enhanced efficiency in quality assurance: The automated evaluation tool could greatly cut down on the time and resources needed for verifying calls. This might enable quality assurance teams to review calls, potentially enhancing the overall service quality and consistency.

2. Standardization of verification process: By pinpointing characteristics of confirmed phone calls the system could aid in streamlining verification procedures among various representatives and call centers. This uniformity might result in enhanced consistency in customer interactions and better adherence to security measures.

3. Real-Time agent support: The current setup is designed for reviewing calls after they end. The model's effectiveness hints at its possible use, in real time scenarios. This could offer agents guidance during calls, ensuring they follow correct verification processes and possibly minimizing mistakes.

4. Training performance management: The model's effectiveness in call analysis hints at its possible use in real time scenarios. This could provide agents with feedback during calls, aiding them in following correct verification protocols and possibly minimizing mistakes.

5. Scalability of quality assurance: Processing several calls efficiently enables better quality control especially as call volumes rise. This adaptability is crucial in meeting the increasing customer service needs and the shift, towards centralized call centers.

6. Cost reduction: By streamlining a part of the verification assessment procedure companies might cut down on expenses linked to manual quality control. This fits in with the industry push to enhance operational efficiency without compromising service excellence.

7. Regulatory compliance: This tool could assist companies in industries with regulations on customer verification, such as financial services in showcasing thorough compliance checks. It has the potential to simplify audits and lessen risks associated with compliance.

8. Customer Experience enhancement: By maintaining consistent adherence to verification procedures this tool has the potential to enhance customer experiences. The implementation of verification not only boosts security but also showcases professionalism and meticulousness elements that play a significant role in fostering customer confidence and contentment.

In summary this research has implications, for both the theoretical comprehension of how machine learning is used in analyzing conversations and the practical side of call center management. It shows how AI could improve quality control procedures potentially resulting in more reliable and secure customer service engagements.

6.4 Limitations

The research has given us insights into using machine learning for call verification evaluation but it's crucial to recognize the limitations. These boundaries help understand the results better and point out directions for studies.

1. Use of synthetic data: One major drawback of this research is the reliance on data to train and assess the model. Although this method enabled testing and sidestepped privacy issues linked to actual customer data it might not encompass the full intricacies and variations found in real life call center exchanges. As pointed

out by pitor and Marta ,synthetic data though beneficial could overlook some subtleties in authentic datasets. [30]

2. Limited linguistic diversity: The research primarily concentrated on phone calls conducted in English. In a world where businesses span across borders call centers frequently handle interactions, in languages. However, the model's effectiveness when dealing with English calls or those involving extensive code switching has not been evaluated yet. This constraint echoes the concerns highlighted by Blodgett and colleagues about the importance of incorporating linguistic diversity in NLP studies. [31]

3. Scope of verification assessment: The current system only looks at whether a call was verified without considering other crucial factors, like resolving issues, customer happiness or following procedural guidelines. This narrow focus, while valuable, provides only a partial view of overall call quality.

4. Potential for overfitting: The model's impressive accuracy on the test set is commendable. It does raise concerns about potential overfitting. Models that excel on test data might have simply memorized specifics rather than grasping generalizable patterns. This limitation emphasizes the importance of validating on real world datasets.

5. Absence of Audio features analysis: The research primarily examined written text. Did not include an assessment of audio aspects like tone, pitch or speech speed which could offer more insights for verification. By focusing on text there is a possibility of overlooking significant nonverbal signals, a constraint often recognized in studies on speech analysis.

6. Limit of test data: The sample dataset consisting of 20 calls although it serves as a validation is considered relatively limited. A broader and more varied test collection would be essential for an assessment of the model's effectiveness and adaptability. This constraint is in line with conversations regarding the significance of testing, in machine learning scenarios.

By recognizing these constraints establish a structure to understand the outcomes of the study and pinpoint areas for further exploration and advancement, in evaluating automated call verification.

6.5 Recommendations for future research

Considering the discoveries and constraints outlined in this study various areas, for investigation have been pinpointed. These suggestions seek to tackle existing constraints, broaden the research reach and advance the utilization of machine learning in call verification evaluation.

1. **Real-World data validation**: Future research should concentrate on verifying the model's effectiveness with call center data allowing for a more accurate evaluation of its performance in practical settings and tackling any constraints linked to artificial data. Following pitor and Marta suggestion, analyzing how the model performs with synthetic data versus real data could offer valuable insights into its overall applicability [30].

2. **Multilingual & Cross-cultural adaptation**: It is essential to enhance the model's capabilities to cater to languages and diverse cultural settings on a global scale. This resonates with the emphasis on embracing linguistic variety in NLP studies as highlighted by Blodgett [31] .Subsequent research endeavors may delve into transfer learning methods to customize the model, for languages and cultural subtleties during verification procedures.

3. **Integration of audio feature analysis**: Using elements like tone, pitch and speech speed in audio could enhance the evaluation of call verification. This combined approach might boost model precision. Offer deeper insights into the verification process.

4. **Comparative analysis with human performance**: Comparing how well the model performs with the work of human quality assurance experts could offer insights into how humans and AI can work together effectively in call centers.

5. **Expansion to measure other call quality assessment :** In the future researchers may consider broadening the models focus to evaluate elements of call quality apart from just verification like resolving issues, questioning techniques ensuring & customer satisfaction.

6. **Ethical AI framework for customer service :** Creating a set of guidelines for using AI in customer service settings with a focus on privacy and consent issues is essential

7. **Real-time feedback system development :** Researching the possibility of using the model as a feedback tool, for call center representatives could greatly boost its real world usefulness. This would require studying fast prediction techniques and finding ways to offer instant feedback without interrupting the call process.

8. **Explainable AI for verification assessment**: Enhancing techniques to improve the transparency and clarity of the models decision making process could boost confidence and usage, in call center environments

Here are some suggestions for studies that seek to overcome existing constraints, broaden the model's relevance, and add value to the wider domain of AI supported quality control, in customer service. By exploring these research paths, can strive for improved efficient automated call verification evaluation systems.

6.6 Summary of a discussion

This study has shown how machine learning and logistic regression can help automate call verification assessment in customer service settings. The model developed displayed accuracy in distinguishing between verified and non-verified calls with potential benefits for enhancing efficiency and consistency in call center quality assurance processes.

The research contributes to both the understanding of NLP applications in conversational analysis and the practical aspects of call center operations. However, there are limitations such as using data and focusing narrowly on verification that point to areas for future exploration.

Suggestions for research include validating real world data adapted for multilingual use incorporating audio features and exploring ethical AI frameworks. These recommendations aim to tackle limitations and broaden the model's usefulness.

In summary this study represents an advancement towards AI supported quality assurance in customer service creating opportunities for more effective, consistent, and secure interactions with customers.

CHAPTER 7: CONCLUSION

7.1 Introduction

This study aimed to create and assess a machine learning system for automated call verification evaluation in customer service settings. The focus was on improving efficiency and reliability in call center quality control procedures specifically emphasizing customer verification. In this section the main discoveries are outlined, the research objectives and questions are revisited, the importance of the outcomes is discussed and the wider impact of this project on AI supported customer service is considered.

7.2 Summary of the research

The research used a combination of methods, blending machine learning methods with natural language processing to examine transcripts of phone calls. A logistic regression model was. Trained using a set of artificial call recordings, which were transcribed and prepared to identify important attributes, for evaluation purposes. The model was then integrated into a user tool tailored for call center quality control teams.

Key stages of the research included:

1.Genaration of synthetic call data
2. Development of custom text vectorization method
3. Training & evaluation of multiple machine learning models
4. Implementation of the best-performing model (logistic regression)
5. Testing & validation of the application using set of 20 diverse call scenario's

7.3 Addressing research questions.

The study successfully addressed the following research questions.

1. **How can NLP techniques be effectively applied to analyze call transcripts for the purpose of customer verification process in call record**? The study showed that using text preparation personalized encoding and logistic regression can accurately evaluate call transcripts, for verification procedures. The impressive level of precision attained indicates that these natural language processing methods can identify the linguistic characteristics associated with accurate verification.

2. **What are the most efficient vectorization methods for converting textual data into numerical form for use in machine learning models?** A specialized method of converting data into vectors, which emphasizes whether key terms are present or absent demonstrated great success in this undertaking. This strategy surpassed approaches such, as TF IDF in accurately representing the core aspects of verification procedures.

3. **How accurately can a linear regression model predict the verification status of call records based on vectorized transcript data?** The logistic regression model performed well on the test dataset achieving a perfect accuracy score of 1.0 and an ROC AUC score of 1.0. This indicates its ability to differentiate between verified and non-verified calls, with high precision.

4. **How can the model's performance be evaluated in terms of accuracy, precision, recall, F1-score & other metrics?** The evaluation of the model involved a range of criteria such as correctness, exactness, completeness, balanced score and the area, under the ROC curve. These measures offered an examination of how well the model performed in various classification scenarios.

5. **What are the potential challenges & limitations in developing this model?** Significant obstacles involved utilizing data, the risk of overfitting and the narrow emphasis solely on verification. These constraints point out avenues for further exploration and advancement in research.

6. **What future improvements can be made to extend the model's capabilities to access additional call quality factors such as hold procedure, warm welcome, end greetings, questioning techniques, emotion detections, clear communication & solution given?** Suggestions for enhancing capabilities involve incorporating analysis of audio features, broadening support, for multiple languages and integrating a comprehensive range of call quality aspects beyond just verification.

7.4 Significance of the findings

1. The outstanding success of the regression model in precisely categorizing confirmed and unconfirmed calls marks a notable progress in automated quality control for call centers.

2. Efficiency improvement : The system that operates automatically can handle a number of phone calls efficiently with the ability to assess all calls rather than just a small subset.

3. .Consistency enhancement : Ensuring that the model applies the standards to all calls guarantees a consistent assessment of verification procedures.

4. Real time potential: The model's effectiveness, though currently used for analyzing calls after they end shows promise, for being applied in time to provide instant feedback during calls.

5. Scalability : The system's capacity to manage high call volumes effectively tackles the scalability issues encountered by expanding call centers.

6. Data-driven & insights : The significance of the model's features can offer insights into the key elements of the authentication process guiding training and enhancements, in procedures.

7.5 Implications & future directions

The effective creation and utilization of this automated phone verification evaluation system hold implications for the customer service industry and the enhancement of quality assurance, with AI support.

1. Shift in Quality Assurance paradigm: This study suggests a scenario in which AI systems can manage regular evaluations enabling human quality assurance experts to concentrate on the intricate and nuanced facets of customer engagements.

2. Enhanced security & compliance: By maintaining verification procedures these systems can greatly contribute to bolstering security measures and upholding regulatory standards, within sectors that deal with confidential customer data.

3. AI-Human collaboration: The system shows how AI and human agents can work together effectively in call centers with technology enhancing judgment instead of replacing it.

4. Ethical AI development: The research highlights the significance of creating AI systems while being mindful of concerns especially when dealing with customer information and impacting interactions, between people.

Further exploration should center on overcoming the constraints of this research by validating the model with actual data, from real life scenarios broadening its reach across different languages and cultures and incorporating a wider range of factors that influence call quality.

7.6 Concluding remarks.

This study has shown how machine learning and NLP methods can greatly improve call center operations, especially when it comes to verifying customers. Though we should approach the flawless results with some caution they do indicate a move towards smoother, reliable, and secure customer service exchanges.

As artificial intelligence advances and becomes more integrated into facets of business activities research such as this sets the foundation for the responsible and efficient deployment of AI supported systems in customer service settings. The future of maintaining quality standards in call centers probably depends on the blend of AIs efficiency and human understanding collaborating to guarantee top notch customer service and security measures.

CHAPTER 8: REFERENCES

[1] salesforce, "State of the connected customer," 2023. [Online]. Available: https://salesforce.com/content/dam/web/en_us/www/documents/research/State-of-the-Connected-Customer.pdf.

[2] Qultiy Assurance & training connection , "QATC Survey Results," [Online]. Available: https://qatc.org/fall-2018-connection/survey-results/. [Accessed 2018].

[3] M. Chui, J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel and S. Malhotra, "Notes from the AI frontier," McKinsey Global Institute., 2018.

[4] T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent trends in deep learning based natural language processing," 2018. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8416973.

[5] IBM, "Cost of data breach report," newyork, 2023.

[6] Live Agent, "Call center benchmarks," 2024. [Online]. Available: https://www.liveagent.com/research/call-center-benchmarks/#:~:text=To%20break%20this%20statistic%20down,the%20industry%20you're%20in..

[7] Grandview Research , "Call Center AI Market Size, Share And Growth Report, 2030," Grandview Research , 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/call-center-artificial-intelligence-market-report#.

[8] T. M. Mitchell, Machine learning., 1997.

[9] Z. Aksin, M. Armony and V. Mehrotra, "The modern call center: A multi-disciplinary perspective on operations management research," Production and Operations Management society, 2007.

[10] E. D. Liddy, "Natural Language Processing," Syracuse University , 2001.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. kaiser and I. Polosukhin, "Attention is all you need. In Advances in neural information processing systems," 2017.

[12] G. Salton and C. Buckley, "Information Processing & Management," Science direct, 1988. [Online]. Available: www.sciencedirect.com/journal/information-processing-and-management/vol/24/issue/5.

[13] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi and i. Lai, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments.," in *Fourth International Conference on Digital Information Processing, Data Mining, and Wireless Communications*, 2012.

[14] V. Venkatesh, M. G. Morris, G. B. Davis and F. D. Davis, "User acceptance of information technology: Toward a unified view.," Management Information Systems Research Center, University of Minnesota, 2003.

[15] P. Voigt and B. v. Dem, The EU general data protection regulation (GDPR). A Practical Guide, Springer International Publishing, 2017.

[16] M. E. Jalal, M. Hosseini and s. karlsson, "Forecasting incoming call volumes in call centers with recurrent," *Journal of Business Research*, vol. 69, pp. 4811-4814, 2016.

[17] S. Ezzart, N. e. Gayar and m. Ghanem, "Sentiment Analysis of Call Centre Audio," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 4, pp. 619-627, 2012.

[18] Y. Gao, T. Parcollet, S. Zaiem, J. Fernandez-Marques, P. P. B. de Gusmao, D. J. Beutel and N. D. Lane, "End-to-End Speech Recognition from Federated Acoustic Models," in *nternational Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.

[19] B. Amirgaliyev, G. Yegemberdiyeva, A. Kuchansky, Y. Andrashko and I. Korol, "Automating the customer verification process in a car sharing system based on machine learning methods," August 2022. [Online]. Available: www.researchgate.net/publication/363423301_Automating_the_customer_verification_process_in_a_car_sharing_system_based_on_machine_learning_methods.

[20] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4738466/authors#authors.

[21] M. A. Jabbar and Suharjito, "Fraud Detection Call Detail Record Using Machine Learning in Telecommunications Company," July 2020. [Online]. Available: www.researchgate.net/publication/343277023_Fraud_Detection_Call_Detail_Record_Using_Machine_Learning_in_Telecommunications_Company.

[22] G. Mishne, d. carmel, R. Hoory, A. Roytman and A. Soffer, "Automatic analysis of call-center conversations," octomber 2005. [Online]. Available: https://www.researchgate.net/publication/221614459_Automatic_analysis_of_call-center_conversations.

[23] N. LI, "Ethical Considerations in Artificial Intelligence," Bank of suzhou, 2023.

[24] P. K Sarma, Y. Liang and W. Sethares, "Domain Adapted Word Embeddings for Improved Sentiment Classification," july 2018. [Online]. Available: https://aclanthology.org/W18-3407.pdf.

[25] p. a. Olujimi and a. ibijola, "NLP techniques for automating responses to customer queries: a systematic review," 15 May 2023. [Online]. Available: https://link.springer.com/article/10.1007/s44163-023-00065-5.

[26] T. Fawcett, " An introduction to ROC analysi," Institute for the Study of Learning and Expertise, 2164 Staunton Court,, Palo Alto, 2006.

[27] A. Przegalinsk, l. Ciechenawoski, A. Stroz, P. Gloor and G. Mazurek, "In bot we trust: A new methodology of chatbot performance measures," *Business Horizons,* vol. 62, no. 6, pp. 785-797, 2019.

[28] T. Dietterich, "Overfitting and Undercomputing inMachine Learning," Department of Computer Science, Oregon State Unllerslty. , Corvallis, 1995.

[29] d. jurafsky and J. H. Martin, "Speech and Language Processing (3rd ed)," 03 February 2024. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[30] p. Boltuc and M. Boltuc, "BICA for AGI," *Cognitive system research* , vol. 62, pp. 57-67, 2020.

[31] S. L. Blodgett, s. Barocas, H. Daumé III and H. Wallach, "Language (Technology) is Power: A Critical Survey of "Bias" in NLP," 28 May 2020. [Online]. Available: https://arxiv.org/abs/2005.14050.

CHAPTER 9 : APPENDICES

Appendix A : Python scripts

A1. 100call30sen20lang.py

```python
import os
import random
from google.cloud import texttospeech
from pydub import AudioSegment

# Set environment variables for Google Cloud credentials
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = r"F:\machine learning\qulity model\service-account-file.json"
os.environ["PATH"] += os.pathsep + r"C:\ProgramData\chocolatey\bin"
# Explicitly set the path to ffmpeg and ffprobe
AudioSegment.converter = r"C:\ProgramData\chocolatey\bin\ffmpeg.exe"
AudioSegment.ffprobe = r"C:\ProgramData\chocolatey\bin\ffprobe.exe"

# Initialize Google Cloud Text-to-Speech client
client = texttospeech.TextToSpeechClient()

def synthesize_speech(text, output_filename, language_code, gender):
    # Set the text input to be synthesized
    synthesis_input = texttospeech.SynthesisInput(text=text)

    # Build the voice request, specify the language code and the ssml voice gender
    voice = texttospeech.VoiceSelectionParams(
        language_code=language_code,
        ssml_gender=gender
    )

    # Select the type of audio file you want returned
    audio_config = texttospeech.AudioConfig(
        audio_encoding=texttospeech.AudioEncoding.MP3
    )

    # Perform the text-to-speech request
    response = client.synthesize_speech(
        input=synthesis_input, voice=voice, audio_config=audio_config
    )

    # Write the response to the output file
    with open(output_filename, "wb") as out:
        out.write(response.audio_content)
```

```
# Define different conversation scenarios
conversation_scenarios = {
    "router_password_change_with_verification": {
        "customer": [
            "Hi, I need help changing the password on my router.",
            "My name is John Doe and my driving license number is ABC123456.",
            "It's a Linksys router.",
            "Yes, I have the app installed.",
            "Okay, I've opened the app.",
            "Alright, I see the option to change the password.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you please verify your identity first? May I have your name and your driving license or passport number?",
            "Thank you, John. Can you tell me what type of router you have?",
            "Great, do you have the Linksys app installed on your phone?",
            "Perfect. Open the app and log in.",
            "Now, go to the settings menu and look for the 'Change Password' option.",
            "Enter your new password and save the changes.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },

    "no_signal_issue_no_verification_no_greetings": {
        "customer": [
            "Hi, I'm not getting any signal on my TV.",
            "Yes, I've checked all the cables.",
            "Yes, the TV is set to the correct input.",
            "Alright, I'll try that.",
            "It worked! The signal is back.",
            "Thank you so much for your help.",
            "Goodbye."
        ],
        "agent": [
            "Have you checked all the cables to make sure they are securely connected?",
            "Is your TV set to the correct input?",
            "Let's try resetting the cable box. Unplug it from the power source, wait 10 seconds, and plug it back in.",
            "Great!",
            "Bye."
        ]
```

```
    },
    "no_signal_issue_with_verification_greetings": {
        "customer": [
            "Hi, I'm not getting any signal on my TV.",
            "My name is Jane Doe and my passport number is XYZ987654.",
            "Yes, I've checked all the cables.",
            "Yes, the TV is set to the correct input.",
            "Alright, I'll try that.",
            "It worked! The signal is back.",
            "Thank you so much for your help.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and
your driving license or passport number?",
            "Thank you, Jane. Have you checked all the cables to make sure they are securely connected?",
            "Is your TV set to the correct input?",
            "Let's try resetting the cable box. Unplug it from the power source, wait 10 seconds, and plug
it back in.",
            "Great! I'm glad to hear the signal is back.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "non_technical_issue_transfer_customer_service": {
        "customer": [
            "Hi, I need help with my billing statement.",
            "Okay, thank you.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry, I can only assist with technical issues. Let me transfer you to our customer
service hotline.",
            "Please hold while I transfer your call.",
            "Goodbye."
        ]
    },
    "internet_slowness_traffic_issue_apology": {
        "customer": [
            "Hi, my internet has been very slow lately.",
            "My name is Mike Smith and my driving license number is DEF654321.",
            "Is there anything I can do about it?",
            "Okay, thank you for letting me know.",
            "Goodbye."
        ],
```

```
        "agent": [
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and
your driving license or passport number?",
            "Thank you, Mike. Let me check our system for any issues in your area.",
            "It seems there is heavy usage traffic on the signal towers in your area. I apologize for the
inconvenience.",
            "Unfortunately, this is a temporary issue. We are working to resolve it as soon as possible.
In the meantime, you might experience some slow speeds during peak hours.",
            "You're welcome. If you have any other questions, feel free to contact us.",
            "Goodbye."
        ]
    },
    "tv_connection_poor_service": {
        "customer": [
            "Hi, my TV connection is not working.",
            "Yes, I have.",
            "Yes, I did.",
            "Can you provide more details or help me troubleshoot further?",
            "Alright, thanks.",
            "Goodbye."
        ],
        "agent": [
            "Have you checked your cables?",
            "Did you try resetting the cable box?",
            "Then I don't know what the issue is. You should call a technician.",
            "No, just call a technician.",
            "Bye."
        ]
    },
    "data_usage_details_no_verification": {
        "customer": [
            "Hi, can you give me my data usage details?",
            "Thank you.",
            "Goodbye."
        ],
        "agent": [
            "Sure, let me check that for you.",
            "You have used 150GB out of your 200GB monthly allowance.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "data_usage_details_with_verification": {
        "customer": [
            "Hi, can you give me my data usage details?",
```

```
            "My name is Anna Lee and my passport number is UVW456789.",
            "Thank you.",
            "Goodbye."
        ],
        "agent": [
            "Sure, can you please verify your identity first? May I have your name and your driving
license or passport number?",
            "Thank you, Anna. Let me check that for you.",
            "You have used 120GB out of your 200GB monthly allowance.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "router_firmware_update_with_verification": {
        "customer": [
            "Hi, I need help updating the firmware on my router.",
            "My name is Sarah Johnson and my passport number is ABC987654.",
            "It's a Netgear router.",
            "Yes, I have the app installed.",
            "Okay, I've opened the app.",
            "Alright, I see the option to update the firmware.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you please verify your identity first? May I have your name
and your driving license or passport number?",
            "Thank you, Sarah. Can you tell me what type of router you have?",
            "Great, do you have the Netgear app installed on your phone?",
            "Perfect. Open the app and log in.",
            "Now, go to the settings menu and look for the 'Update Firmware' option.",
            "Follow the instructions to update the firmware and wait for it to complete.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "router_firmware_update_without_verification": {
        "customer": [
            "Hi, I need help updating the firmware on my router.",
            "It's a Netgear router.",
            "Yes, I have the app installed.",
            "Okay, I've opened the app.",
            "Alright, I see the option to update the firmware.",
            "Thank you for your help!",
            "Goodbye."
```

```
        ],
        "agent": [
            "Sure, I can help with that. Can you tell me what type of router you have?",
            "Great, do you have the Netgear app installed on your phone?",
            "Perfect. Open the app and log in.",
            "Now, go to the settings menu and look for the 'Update Firmware' option.",
            "Follow the instructions to update the firmware and wait for it to complete.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "no_internet_connection_no_verification_no_greetings": {
        "customer": [
            "Hi, I have no internet connection at all.",
            "Yes, I did.",
            "No, it's off.",
            "I've checked them, they seem fine.",
            "Alright, I'll try that.",
            "It's working now, the internet is back.",
            "Thank you for your help.",
            "Goodbye."
        ],
        "agent": [
            "Have you tried restarting your router?",
            "Is the internet light on your router blinking?",
            "Let's check the cables to ensure they are securely connected.",
            "Can you unplug the router from the power source, wait 10 seconds, and plug it back in?",
            "Great.",
            "Bye."
        ]
    },
    "no_internet_connection_with_verification_greetings": {
        "customer": [
            "Hi, I have no internet connection at all.",
            "My name is Emily Davis and my driving license number is XYZ123456.",
            "Yes, I did.",
            "No, it's off.",
            "I've checked them, they seem fine.",
            "Alright, I'll try that.",
            "It's working now, the internet is back.",
            "Thank you for your help.",
            "Goodbye."
        ],
        "agent": [
```

```
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and
your driving license or passport number?",
            "Thank you, Emily. Have you tried restarting your router?",
            "Is the internet light on your router blinking?",
            "Let's check the cables to ensure they are securely connected.",
            "Can you unplug the router from the power source, wait 10 seconds, and plug it back in?",
            "Great! I'm glad to hear the internet is back.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "billing_issue_transfer_customer_service": {
        "customer": [
            "Hi, I have a question about a charge on my bill.",
            "Okay, thank you.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry, I can only assist with technical issues. Let me transfer you to our customer
service hotline.",
            "Please hold while I transfer your call.",
            "Goodbye."
        ]
    },
    "internet_slowness_due_to_maintenance_apology": {
        "customer": [
            "Hi, my internet has been very slow lately.",
            "My name is Tom Harris and my passport number is ABC654321.",
            "How long will this maintenance take?",
            "Okay, thank you for letting me know.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and
your driving license or passport number?",
            "Thank you, Tom. Let me check our system for any issues in your area.",
            "It seems there is scheduled maintenance in your area. I apologize for the inconvenience.",
            "It should be completed within the next few hours. You might experience slow speeds until
then.",
            "You're welcome. If you have any other questions, feel free to contact us.",
            "Goodbye."
        ]
    },
    "tv_connection_poor_service": {
        "customer": [
```

```
            "Hi, my TV connection is not working.",
            "Yes, I have.",
            "Yes, I did.",
            "Can you provide more details or help me troubleshoot further?",
            "Alright, thanks.",
            "Goodbye."
        ],
        "agent": [
            "Have you checked your cables?",
            "Did you try resetting the cable box?",
            "Then I don't know what the issue is. You should call a technician.",
            "No, just call a technician.",
            "Bye."
        ]
    },
    "data_usage_details_without_verification": {
        "customer": [
            "Hi, can you give me my data usage details?",
            "Thank you.",
            "Goodbye."
        ],
        "agent": [
            "Sure, let me check that for you.",
            "You have used 75GB out of your 100GB monthly allowance.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "data_usage_details_with_verification": {
        "customer": [
            "Hi, can you give me my data usage details?",
            "My name is Lisa Brown and my driving license number is DEF789012.",
            "Thank you.",
            "Goodbye."
        ],
        "agent": [
            "Sure, can you please verify your identity first? May I have your name and your driving
license or passport number?",
            "Thank you, Lisa. Let me check that for you.",
            "You have used 200GB out of your 300GB monthly allowance.",
            "You're welcome. Have a great day!",
            "Goodbye."
        ]
    },
    "slow_internet_due_to_device_interference": {
```

```
        "customer": [
            "Hi, my internet speed is very slow.",
            "My name is Mark Wilson and my passport number is GHI321654.",
            "Is there anything I can do about it?",
            "Okay, I'll try that. Thank you for your help.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and
your driving license or passport number?",
            "Thank you, Mark. Let me check our system for any issues in your area.",
            "It seems there is interference from other devices in your area causing the slow speed. I
apologize for the inconvenience.",
            "You can try moving your router to a different location or reduce the number of connected
devices.",
            "You're welcome. If you have any other questions, feel free to contact us.",
            "Goodbye."
        ]
    },
    "router_port_forwarding_setup_with_verification": {
        "customer": [
            "Hi, I need help setting up port forwarding on my router.",
            "My name is Robert Green and my passport number is XYZ654123.",
            "It's a TP-Link router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the port forwarding option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you please verify your identity first? May I have your name
and your driving license or passport number?",
            "Thank you, Robert. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'Port Forwarding' option and click on it.",
            "Enter the required details and save the settings.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "router_port_forwarding_setup_without_verification": {
        "customer": [
            "Hi, I need help setting up port forwarding on my router.",
```

```
            "It's a TP-Link router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the port forwarding option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'Port Forwarding' option and click on it.",
            "Enter the required details and save the settings.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "satellite_tv_signal_weak_no_verification": {
        "customer": [
            "Hi, my satellite TV signal is very weak.",
            "Yes, I have checked the dish alignment.",
            "No, the weather is clear.",
            "Okay, I'll try resetting the receiver.",
            "It seems to be better now.",
            "Thank you for your help.",
            "Goodbye."
        ],
        "agent": [
            "Have you checked the dish alignment?",
            "Is there any bad weather that could be affecting the signal?",
            "Let's try resetting the receiver. Unplug it from the power source, wait 10 seconds, and plug
it back in.",
            "Great! I'm glad to hear it's better.",
            "You're welcome.",
            "Goodbye."
        ]
    },
    "satellite_tv_signal_weak_with_verification": {
        "customer": [
            "Hi, my satellite TV signal is very weak.",
            "My name is Jessica White and my driving license number is LMN123456.",
            "Yes, I have checked the dish alignment.",
            "No, the weather is clear.",
            "Okay, I'll try resetting the receiver.",
            "It seems to be better now.",
```

```
            "Thank you for your help.",
            "Goodbye."
        ],
        "agent": [
            "I'm sorry to hear that. Can you please verify your identity first? May I have your name and your driving license or passport number?",
            "Thank you, Jessica. Have you checked the dish alignment?",
            "Is there any bad weather that could be affecting the signal?",
            "Let's try resetting the receiver. Unplug it from the power source, wait 10 seconds, and plug it back in.",
            "Great! I'm glad to hear it's better.",
            "You're welcome.",
            "Goodbye."
        ]
    },
    "router_factory_reset_without_verification": {
        "customer": [
            "Hi, I need help resetting my router to factory settings.",
            "It's a D-Link router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the factory reset option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'Factory Reset' option and click on it.",
            "Confirm the reset and wait for the router to reboot.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "router_factory_reset_with_verification": {
        "customer": [
            "Hi, I need help resetting my router to factory settings.",
            "My name is Andrew Brown and my driving license number is GHI987654.",
            "It's a D-Link router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the factory reset option.",
            "Thank you for your help!",
            "Goodbye."
```

```
        ],
        "agent": [
            "Sure, I can help with that. Can you please verify your identity first? May I have your name and your driving license or passport number?",
            "Thank you, Andrew. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'Factory Reset' option and click on it.",
            "Confirm the reset and wait for the router to reboot.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "satellite_tv_box_software_update_without_verification": {
        "customer": [
            "Hi, I need help updating the software on my satellite TV box.",
            "It's a DirectTV box.",
            "Yes, I have the remote.",
            "Okay, I've accessed the settings menu.",
            "I see the software update option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you tell me what type of satellite TV box you have?",
            "Great, do you have the remote with you?",
            "Please use the remote to access the settings menu.",
            "Now, look for the 'Software Update' option and select it.",
            "Follow the instructions to update the software and wait for it to complete.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "satellite_tv_box_software_update_with_verification": {
        "customer": [
            "Hi, I need help updating the software on my satellite TV box.",
            "My name is Oliver Smith and my passport number is JKL123987.",
            "It's a DirectTV box.",
            "Yes, I have the remote.",
            "Okay, I've accessed the settings menu.",
            "I see the software update option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
```

```
            "Sure, I can help with that. Can you please verify your identity first? May I have your name
and your driving license or passport number?",
            "Thank you, Oliver. Can you tell me what type of satellite TV box you have?",
            "Great, do you have the remote with you?",
            "Please use the remote to access the settings menu.",
            "Now, look for the 'Software Update' option and select it.",
            "Follow the instructions to update the software and wait for it to complete.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "router_dns_settings_change_without_verification": {
        "customer": [
            "Hi, I need help changing the DNS settings on my router.",
            "It's a Belkin router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the DNS settings option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'DNS Settings' option and click on it.",
            "Enter the new DNS server addresses and save the settings.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    },
    "router_dns_settings_change_with_verification": {
        "customer": [
            "Hi, I need help changing the DNS settings on my router.",
            "My name is Laura Thompson and my driving license number is QRS987321.",
            "It's a Belkin router.",
            "Yes, I can access the router settings page.",
            "Okay, I've logged in.",
            "I see the DNS settings option.",
            "Thank you for your help!",
            "Goodbye."
        ],
        "agent": [
            "Sure, I can help with that. Can you please verify your identity first? May I have your name
and your driving license or passport number?",
```

```
            "Thank you, Laura. Can you tell me what type of router you have?",
            "Great, do you have access to the router settings page?",
            "Please log in and go to the settings menu.",
            "Now, look for the 'DNS Settings' option and click on it.",
            "Enter the new DNS server addresses and save the settings.",
            "You're welcome! Have a nice day.",
            "Goodbye."
        ]
    }
}

# Define a list of voice configurations
voices = [
    {"language_code": "en-US", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-US", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-GB", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-GB", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-AU", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-AU", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-IN", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-IN", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-CA", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-CA", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-NZ", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-NZ", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-IE", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-IE", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
    {"language_code": "en-ZA", "ssml_gender": texttospeech.SsmlVoiceGender.MALE},
    {"language_code": "en-ZA", "ssml_gender": texttospeech.SsmlVoiceGender.FEMALE},
]

# Create a folder to store the audio files
audio_folder = r"F:\machine learning\qulity model\audio"
if not os.path.exists(audio_folder):
    os.makedirs(audio_folder)

# Generate 100 sample calls
for call_index in range(100):
    # Randomly select a conversation scenario
    scenario_key = random.choice(list(conversation_scenarios.keys()))
    scenario = conversation_scenarios[scenario_key]

    # Randomly select voices for customer and agent
    customer_voice = random.choice(voices)
    agent_voice = random.choice(voices)
```

```
    # Generate customer audio files
    for i, text in enumerate(scenario["customer"]):
        output_filename = os.path.join(audio_folder, f"call_{call_index}_customer_{i}.mp3")
        synthesize_speech(text, output_filename, customer_voice["language_code"],
customer_voice["ssml_gender"])

    # Generate agent audio files
    for i, text in enumerate(scenario["agent"]):
        output_filename = os.path.join(audio_folder, f"call_{call_index}_agent_{i}.mp3")
        synthesize_speech(text, output_filename, agent_voice["language_code"], agent_voice["ssml_gender"])

    # Combine the audio files
    combined_audio = AudioSegment.empty()

    # Load and combine the audio segments
    for i in range(len(scenario["customer"])):
        customer_audio = AudioSegment.from_mp3(os.path.join(audio_folder,
f"call_{call_index}_customer_{i}.mp3"))
        combined_audio += customer_audio
        if i < len(scenario["agent"]):
            agent_audio = AudioSegment.from_mp3(os.path.join(audio_folder,
f"call_{call_index}_agent_{i}.mp3"))
            combined_audio += agent_audio

    # Specify the custom file location for the combined file
    combined_file_location = os.path.join(audio_folder, f"combined_call_{call_index}.mp3")

    # Export the combined audio to the specified file location
    combined_audio.export(combined_file_location, format="wav")

print("Generated 100 sample calls.")
```

A.2 audioanlyz.py

```
import whisper
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pandas as pd
import os
import time
import torch

# Check for GPU availability
```

```
device = "cuda" if torch.cuda.is_available() else "cpu"

# Load the Whisper model
model = whisper.load_model("base", device=device)

# Function to preprocess text
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    words = word_tokenize(text)
    words = [word for word in words if word not in stopwords.words('english')]
    return ' '.join(words)

# Function to transcribe audio files
def transcribe_audio(file_path):
    try:
        result = model.transcribe(file_path)
        return result["text"]
    except Exception as e:
        print(f"Error transcribing file {file_path}: {e}")
        return ""

# Load the CSV file
file_path = r'F:\machine learning\qulity model\test model\test.csv'
data = pd.read_csv(file_path)

# Assuming the audio files are named with their Call ID and located in a directory named 'audio_files'
audio_dir = r'F:\machine learning\qulity model\test model\test calls wav'  # Update this with the actual
path to your audio files
for index, row in data.iterrows():
    data.at[index, 'Transcription'] = transcribe_audio(os.path.join(audio_dir, f"{row['Call ID']}.wav"))
    time.sleep(2)  # Introduce a short break between each transcription to reduce CPU usage and heat

# Preprocess the transcriptions
data['Cleaned Transcription'] = data['Transcription'].apply(preprocess_text)

# Save the data with transcriptions to the specified location
output_file_path = r'F:\machine learning\qulity model\test model\model 3\transcribed_data.csv'
data.to_csv(output_file_path, index=False)
```

A3. model_&_vector_build.py (for model training & evaluation)

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from sklearn.metrics import classification_report, roc_auc_score, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
import dill as pickle

# Load the CSV file with transcriptions
file_path = r'F:\machine learning\qulity model\test model\model
3\transcribed_data.csv'
data = pd.read_csv(file_path)
print(data.shape)
print(data['Cleaned Transcription'].head(5))

# Vocabulary
vocab = Counter()
for sentence in data['Cleaned Transcription']:
    vocab.update(sentence.split())

print("vocabulary size:",len(vocab))

tokens = [key for key in vocab if vocab[key] > 15]

print("Number of tokens appearing more than 15 times:", len(tokens))
print("Most common tokens:", tokens[:10])

# Save tokens to a text file
tokens_file_path = r'F:\machine learning\qulity model\test model\model 3\test
runs\tokens.txt'
with open(tokens_file_path, 'w') as file:
    for token in tokens:
        file.write(f"{token}\n")

# Divide data set
X = data['Cleaned Transcription']
```

```
y = data['Verifications']
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

print("Training set size:",X_train.shape)
print("Testing set size:",X_test.shape)
print(y_train)

# Extract features using a custom vectorizer
def vectorizer(data, tokens):
    vectorize_lst = []

    for sentence in data:
        sentence_vec = np.zeros(len(tokens))

        for i in range(len(tokens)):
            if tokens[i] in sentence.split():
                sentence_vec[i] = 1
        vectorize_lst.append(sentence_vec)

    vectorize_lst_np = np.asarray(vectorize_lst, dtype=np.float32)

    return vectorize_lst_np

vectorized_X_train = vectorizer(X_train, tokens)
vectorized_X_test = vectorizer(X_test, tokens)

print(vectorized_X_test)

# Define the plot_distribution function
def plot_distribution(y, title):
    value_counts = y.value_counts()
    unique_values = value_counts.index.tolist()
    counts = value_counts.values

    print(f"{title}:")
    print("Unique values:", unique_values)
    print("Counts:", counts)

    if len(unique_values) >= 2:
        plt.figure(figsize=(8, 6))
        plt.pie(counts, labels=unique_values, autopct='%1.1f%%')
        plt.title(title)
        plt.show()
```

```
    else:
        print(f"Not enough unique values to create a pie chart for {title}.")

#distribution of call trancript lenght
data['text_length'] = data['Cleaned Transcription'].apply(lambda x:
len(x.split()))
print("text data;",data['text_length'].describe())
plt.figure(figsize=(10, 6))
plt.hist(data['text_length'], bins=30)
plt.title('Distribution of Call Transcript Lengths')
plt.xlabel('Word Count')
plt.ylabel('Frequency')
plt.show()

#Handle imbalanced data
print("Original data distribution:")
print(y_train.value_counts())
plot_distribution(y_train, "Original Data Distribution")

smote = SMOTE()
vectorized_X_train_smote, y_train_smote = smote.fit_resample(vectorized_X_train,
y_train)

print("Data distribution after SMOTE:")
print(y_train_smote.value_counts())
plot_distribution(y_train_smote, "Data Distribution After SMOTE")

# Define a function to train, predict, and evaluate a model
def train_predict_evaluate(model, model_name, X_train, y_train, X_test, y_test):
    print(f"\n--- {model_name} ---")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    try:
        y_pred_proba = model.predict_proba(X_test)[:, 1]
    except AttributeError:
        # If the model doesn't have predict_proba, use decision_function if
available
        try:
            y_pred_proba = model.decision_function(X_test)
        except AttributeError:
            y_pred_proba = y_pred  # Fall back to binary predictions

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy}")
```

```
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    try:
        roc_auc = roc_auc_score(y_test, y_pred_proba)
        print(f"ROC-AUC Score: {roc_auc}")
    except ValueError:
        print("ROC-AUC Score could not be calculated.")
        roc_auc = None

    return model, y_pred, accuracy, roc_auc

# Initialize models
models = {
    "Logistic Regression": LogisticRegression(),
    "K-Nearest Neighbors": KNeighborsClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC(probability=True),
    "Naive Bayes": GaussianNB()
}

# Train, predict, and evaluate each model
results = {}
accuracies = []
roc_auc_scores = []
for model_name, model in models.items():
    trained_model, y_pred, accuracy, roc_auc = train_predict_evaluate(model,
model_name, vectorized_X_train_smote, y_train_smote, vectorized_X_test, y_test)
    results[model_name] = {
        'model': trained_model,
        'predictions': y_pred,
        'accuracy': accuracy,
        'roc_auc': roc_auc
    }
    accuracies.append(accuracy)
    roc_auc_scores.append(roc_auc if roc_auc is not None else 0)

# Create a DataFrame to include Call ID, Actual values, and Predicted values for
each model
results_df = pd.DataFrame({
    'Call ID': data.loc[y_test.index, 'Call ID'],
    'Actual': y_test.values
})

for model_name, result in results.items():
```

```
    results_df[f'Predicted_{model_name}'] = result['predictions']

# Print the results DataFrame
print("\nPrediction Results:")
print(results_df)

# Save results to CSV
results_df.to_csv(r'F:\machine learning\qulity model\test model\model 3\test
runs\multi_model_results.csv', index=False)

# Save only the Logistic Regression model
logistic_model = results['Logistic Regression']['model']
model_file_path = r'F:\machine learning\qulity model\test model\model 3\test
runs\logistic_regression_model.pkl'
with open(model_file_path, 'wb') as model_file:
    pickle.dump(logistic_model, model_file)
print("\nLogistic Regression model has been saved.")

# Save the vectorizer to a file
vectorizer_file_path = r'F:\machine learning\qulity model\test model\model 3\test
runs\tfidf_vectorizer.pkl'
with open(vectorizer_file_path, 'wb') as vectorizer_file:
    pickle.dump(vectorizer, vectorizer_file)
print("\nvectorizer has been saved.")

# Create bar plots for accuracy and ROC-AUC scores
def create_bar_plot(scores, title, ylabel):
    plt.figure(figsize=(10, 6))
    bars = plt.bar(models.keys(), scores)
    plt.title(title)
    plt.xlabel('Models')
    plt.ylabel(ylabel)
    plt.xticks(rotation=45, ha='right')

    # Add value labels on top of each bar
    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2., height,
                 f'{height:.2f}',
                 ha='center', va='bottom')

    plt.tight_layout()
    plt.show()

create_bar_plot(accuracies, 'Model Accuracies', 'Accuracy')
```

```
create_bar_plot(roc_auc_scores, 'Model ROC-AUC Scores', 'ROC-AUC Score')

print("\nAll models have been trained and evaluated. Graphs have been created for accuracies and ROC-AUC scores.")
```

A4. finalcodegui2.1.py (final app code)

```
mport whisper
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pandas as pd
import os
import time
import torch
import tkinter as tk
from tkinter import filedialog, messagebox
import dill as pickle  # Use dill instead of pickle for serialization
import numpy as np

# Check for GPU availability
device = "cuda" if torch.cuda.is_available() else "cpu"

# Load the Whisper model
whmodel = whisper.load_model("base", device=device)

# Ensure NLTK stopwords are downloaded
import nltk
nltk.download('stopwords')
nltk.download('punkt')

# Function to preprocess text
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    words = word_tokenize(text)
    words = [word for word in words if word not in stopwords.words('english')]
    return ' '.join(words)

# Function to transcribe audio files
def transcribe_audio(file_path):
    try:
        result = whmodel.transcribe(file_path)
        return result["text"]
    except Exception as e:
        print(f"Error transcribing file {file_path}: {e}")
```

```
        return ""

# Function to process a list of audio files provided by the user
def process_audio_files(audio_files):
    rows = []

    for audio_file in audio_files:
        transcription = transcribe_audio(audio_file)
        cleaned_transcription = preprocess_text(transcription)
        rows.append({'Audio File': os.path.basename(audio_file), 'Cleaned
Transcription': cleaned_transcription})
        time.sleep(2)  # Introduce a short break between each transcription to
reduce CPU usage and heat

    data = pd.DataFrame(rows)
    return data

# Load logistic regression model and vectorizer & tokens
def load_model_and_vectorizer_tokens(model_path, vectorizer_path, tokens_path):
    with open(model_path, 'rb') as model_file:
        loaded_model = pickle.load(model_file)
    with open(vectorizer_path, 'rb') as vectorizer_file:
        loaded_vectorizer = pickle.load(vectorizer_file)
    with open(tokens_path, 'rb') as tokens_file:
        loaded_tokens = pickle.load(tokens_file)
    print("Tokens loaded successfully")
    # Debugging step: check the type of vectorizer
    print(f"Loaded vectorizer type: {type(loaded_vectorizer)}")
    print(f"Loaded model type: {type(loaded_model)}")
    return loaded_model, loaded_vectorizer, loaded_tokens

# Function to predict verification status
def predict_verification(text, model, vectorizer, tokens):
    text_vector = vectorizer([text], tokens)
    prediction = model.predict(text_vector)
    return prediction[0]

# GUI Application
class TranscriptionApp:
    def __init__(self, root, model, vectorizer, tokens):
        self.root = root
        self.root.title("Audio Transcription and Verification App")

        self.audio_files = []
        self.model = model
```

```
        self.vectorizer = vectorizer
        self.tokens = tokens

        self.file_label = tk.Label(root, text="No audio files selected.")
        self.file_label.pack()

        self.add_button = tk.Button(root, text="Add Audio Files",
command=self.add_files)
        self.add_button.pack()

        self.output_button = tk.Button(root, text="Select Output File",
command=self.select_output_file)
        self.output_button.pack()

        self.transcribe_button = tk.Button(root, text="Transcribe and Verify",
command=self.transcribe_and_verify)
        self.transcribe_button.pack()

        self.output_file = None

    def add_files(self):
        files = filedialog.askopenfilenames(filetypes=[("Audio Files", "*.wav
*.mp3")])
        if files:
            self.audio_files.extend(files)
            self.file_label.config(text=f"{len(self.audio_files)} files
selected.")

    def select_output_file(self):
        self.output_file = filedialog.asksaveasfilename(defaultextension=".csv",
filetypes=[("CSV Files", "*.csv")])
        if self.output_file:
            messagebox.showinfo("Selected Output File", f"Output file:
{self.output_file}")

    def transcribe_and_verify(self):
        if not self.audio_files:
            messagebox.showwarning("No Files", "Please add audio files to
transcribe.")
            return

        if not self.output_file:
            messagebox.showwarning("No Output File", "Please select an output
file.")
            return
```

```
        results = process_audio_files(self.audio_files)
        results['Verification'] = results['Cleaned Transcription'].apply(
            lambda x: predict_verification(x, self.model, self.vectorizer,
self.tokens)
        )
        results.to_csv(self.output_file, index=False)
        messagebox.showinfo("Transcription and Verification Completed", f"Results
saved to {self.output_file}")

def main():
    model_path = r"F:\machine learning\qulity model\test model\model 3\software
dev\callcheck_model.pkl"
    vectorizer_path = r"F:\machine learning\qulity model\test model\model
3\software dev\token_vect.pkl"
    tokens_path = r'F:\machine learning\qulity model\test model\model 3\software
dev\tokens.pkl'

    model, vectorizer, tokens = load_model_and_vectorizer_tokens(model_path,
vectorizer_path, tokens_path)

    root = tk.Tk()
    app = TranscriptionApp(root, model, vectorizer, tokens)
    root.mainloop()

if __name__ == "__main__":
    main()
```