

MINOR PROJECT REPORT
ON
“Sentiment Analysis of incoming calls on helpdesk”

**A MINOR PROJECT REPORT SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE OF**

BACHELOR OF ENGINEERING
In
Computer Science & Engineering

SUBMITTED BY

Sukrit Upadhaya (2020a1r069)
Bhanu Partap Singh(2020a1r070)
Osheen Pandita(2020a1t081)
Ketan Arora (2020a1r090)



AUTONOMOUS

SUBMITTED TO
Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Jammu, India
2022

CANDIDATES' DECLARATION

We, **Bhanu Partap Singh, Roll Number:2020a1r070, Sukrit Upadhaya, Roll Number: 2020a1r069, Osheen Pandita, Roll Number:2020a1t081 ,and Ketan Arora, Roll Number: 2020a1r090**, hereby declare that the work which is being presented in the minor project report entitled, “**Sentiment Analysis of incoming calls on helpdesk**” in the partial fulfillment of requirement for the award of degree of B.E. (CSE) and submitted in the Computer Science and Engineering Department, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of our own work carried by us under the supervision of **Ms. Shafalika Vijayal** (Asst. Professor C.S.E, Model Institute of Engineering and Technology, Jammu).The matter presented in this seminar report has not been submitted in this or any other University/Institute for the award of B.E. Degree.

Signature of the students

Dated: 10-12-2022

Sukrit Upadhaya (2020a1r069)
Bhanu Partap Singh(2020a1r070)
Osheen Pandita(2020a1t081)
Ketan Arora (2020a1r090)

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Kot Bhalwal, Jammu, India
(NAAC “A” Grade Accredited)

Ref. No.: 2020A1R070

Date: 25th November 2022

CERTIFICATE

Certified that this seminar report entitled “**Sentiment Analysis of incoming calls on helpdesk**” is the bonafide work of “**Bhanu Partap Singh, Roll Number:2020a1r070, Sukrit Upadhaya, Roll Number: 2020a1r069, Osheen Pandita, Roll Number:2020a1t081 ,and Ketan Arora, Roll Number: 2020a1r090, of 7th Semester, CSE, Model Institute of Engineering and Technology (Autonomous), Jammu**”, who carried out the minor project work under my supervision during September, 2023

Miss Shafalika Vijayal
Mentor-Internal Supervisor
Assistant Professor
Department of CSE, MIET

This is to certify that the above statement is correct to the best of my knowledge.

Ms. Shafalika Vijayal
Program Manager
Computer Science and Engineering, MIET

ACKNOWLEDGEMENTS

Minor projects serve as vital platforms for students to delve into updated technologies and gain firsthand experience in the dynamic field of engineering. The culmination of our collaborative effort has shaped the present work, and I extend my sincere appreciation to all those who have contributed to its success.

I am deeply grateful to **Prof. (Dr.) Ankur Gupta, Director of MIET, Prof. (Dr.) Ashok Kumar, Dean Academics at MIET, and Ms. Shafalika Vijayal, Assistant Professor & Program Manager CSE at MIET.** Their unwavering guidance, constant inspiration, and encouragement were pivotal to the completion of this project. Their keen involvement throughout the entire duration of our work has been truly valuable.

Reflecting on our journey, I extend heartfelt thanks to each member of our team. Together, we navigated through challenges, shared ideas, and collectively worked towards achieving our goals. The commitment and hard work of every team member have been indispensable.

In expressing gratitude, I also acknowledge the teachers who, despite their busy schedules, provided us with their invaluable time, guidance, and support. Their assistance allowed us to carry out our project within the esteemed organization and enriched our training experience. This opportunity marks a significant milestone in our collective career development.

Our team is sincerely thankful to Model Institute of Engineering and Technology (Autonomous), Jammu for providing us with this valuable opportunity. We are committed to applying the skills and knowledge gained during this project in the most effective manner. As a team, we look forward to future collaborations and endeavors.

Bhanu Partap Singh(2020a1r070)

Sukrit Upadhaya(2020a1r069)

Ketan Arora(2020a1r090)

Osheen Pandita(2020a1t081)

ABSTRACT

Call Centers or Support Centers in different companies aggregate huge amount of data everyday. From all the conversations, few conversations are not customer satisfactory i.e sometimes the customer is not satisfied with the support. Finding the sentiment of the customer helps in determining whether the customer was satisfied with the service.

We in this project aim at separating the sources from the conversation and classifying the sentiment of every chunk (speaker 1 and speaker 2) of the audio. The Python script presents a comprehensive solution for real-time speech-to-text conversion and sentiment analysis. Leveraging the Speech Recognition library, the system captures and transcribes spoken words, addressing the need for prompt and accurate transcription services. Additionally, Natural Language Processing (NLP) techniques, facilitated by the NLTK library, are employed for tasks like tokenization, lemmatization, and sentiment analysis. The integration of these components enables a user-friendly interface, allowing users to upload audio files for analysis.

The technology stack comprises Python as the programming language, Speech Recognition for speech processing, NLTK for NLP tasks, and Matplotlib for visualization. The architecture involves a seamless flow from user interaction through the user interface to real-time speech recognition, NLP processing, and sentiment analysis. The system's objective is to enhance accessibility and user experience by providing a robust solution for analyzing spoken content and extracting sentiment insights.

Contents

Candidates' Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Contents	v
List of Figures	vii
Chapter 1 Python	1-3
1.1 Introduction to Python	1
1.2 History of Python	1
1.3 Development in Python	2
1.4 Features of Python	2
1.5 Use of Python	3
Chapter 2 Artificial Intelligence	4-8
2.1 About Artificial Intelligence	11
2.2 History of Artificial Intelligence	11
2.3 Types of Artificial Intelligence	15
2.4 Applications of Artificial Intelligence	15
Chapter 3 Machine Learning	9-14
3.1 About Machine Learning	9
3.2 Difference - Human Learning & Machine Learning	9
3.3 Difference - Rule Based Approach & Machine Learning	10
3.4 Problems solved using Machine Learning	10
3.5 Types of Machine Learning	13
3.6 Procedure of Machine Learning	14
Chapter 4 Introduction	15-17
4.1 Understanding Sentiment Analysis	15
4.2 Applying Sentiment Analysis to Incoming Calls	15
4.3 Benefits for Help Desks	16
4.4 Process Overview	
4.5 Challenges and Considerations	17

Chapter 5 Implementation Highlights	18-23
5.1 Data Collection	18
5.2 Data Cleaning	19
5.2.1 Advantages of Data Cleaning	20
5.3 Data Analysis and Exploration	21
5.3.1 Understanding the Data Landscape	21
5.3.2 Exploratory Data Analysis (EDA)	21
5.3.3 Linguistic and Semantic Analysis	21
5.3.4 Temporal Analysis	22
5.3.5 User-Level Analysis	22
5.3.6 Addressing Imbalances	22
5.3.7 Feature Engineering	22
5.3.8 Preprocessing and Cleaning	22
5.3.9 Validation and Model Iteration	23
 Chapter 6 Modules & Libraries	 24-26
6.1 Python Modules	24
6.2 Python Libraries	25-26
6.2.1 Audio Processing and Speech-to-Text:	
6.2.2 Text Preprocessing:	
6.2.3 Sentiment Analysis:	
6.2.4 Additional Libraries	
 Chapter 7 Project Description	 27-29
7.1 Problem Statement	27
7.2 Workflow of Project	28
7.3 Significance of Important Code Segments	29
Conclusion	30-31
References	32

List of Figures

Figure Number	Figure Title	Page Number
2.1	Applications of AI	4
2.2	Concepts related to AI	4
2.3	History of AI	5
2.4	Application fields of AI	8
3.1	Human Learning	9
3.2	Machine Learning	9
3.3	Rule Based Approach	10
3.4	Machine Learning Approach	10
3.5	Supervised Machine Learning	11
3.6	Unsupervised Machine Learning	12
3.7	Unsupervised Machine Learning - Clustering	12
3.8	Semi-supervised Machine Learning	13
3.9	Reinforcement Learning.	13
3.10	Machine Learning Process	14
4.1	General Sentiments	15
7.1	Workflow	27
7.2	Model Definition	28
7.3	Model Compilation	29
7.4	Sentiment Analysis Graph	29

Chapter 1

Python

Python is the gift that keeps on giving.

The more you understand Python, the more you can do in the 21st Century. As simple as that.

1.1 Introduction to Python

Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming. It's everywhere, and people use numerous Python-powered devices on a daily basis, whether they realize it or not.

1.2 History of Python

Python was created by Guido van Rossum, and first released on February 20, 1991. While you may know the python as a large snake, the name of the Python programming language comes from an old BBC television comedy sketch series called Monty Python's Flying Circus.

One of the amazing features of Python is the fact that it is actually one person's work. Usually, new programming languages are developed and published by large companies employing lots of professionals, and due to copyright rules, it is very hard to name any of the people involved in the project. Python is an exception.

Of course, Guido van Rossum did not develop and evolve all the Python components himself. The speed with which Python has spread around the world is a result of the continuous work of thousands (very often anonymous) programmers, testers, users (many of them aren't IT specialists) and enthusiasts, but it must be said that the very first idea (the seed from which Python sprouted) came to one head – Guido's.

1.3 Development in Python

Python is maintained by the Python Software Foundation, a non-profit membership organization and a community devoted to developing, improving, expanding, and popularizing the Python language and its environment.

1.4 Features of Python

Python is omnipresent, and people use numerous Python-powered devices on a daily basis, whether they realize it or not. There are billions of lines of code written in Python, which means almost unlimited opportunities for code reuse and learning from well-crafted examples. What's more, there is a large and very active Python community, always happy to help.

There are also a couple of factors that make Python great for learning:

- It is easy to learn – the time needed to learn Python is shorter than for many other languages; this means that it's possible to start the actual programming faster;
- It is easy to use for writing new software – it's often possible to write code faster when using Python;
- It is easy to obtain, install and deploy – Python is free, open and multiplatform; not all languages can boast that.

1.5 Use of Python

Programming skills prepare you for careers in almost any industry, and are required if you want to continue to more advanced and higher-paying software development and engineering roles. Python is the programming language that opens more doors than any other. With a solid knowledge of Python, you can work in a multitude of jobs and a multitude of industries. And the more you understand Python, the more you can do in the 21st Century. Even if you don't need it for work, you will find it useful to know.

Many developing tools are implemented in Python. More and more everyday use applications are being written in Python. Lots of scientists have abandoned expensive proprietary tools and switched to Python. Lots of IT project testers have started using Python to carry out repeatable test procedures. The list is long.

Python is a great choice for:

- Web and Internet development (e.g., Django and Pyramid frameworks, Flask and Bottle micro-frameworks)
- Scientific and numeric computing (e.g., SciPy – a collection of packages for the

purposes of mathematics, science, and engineering; Ipython – an interactive shell that features editing and recording of work sessions)

- Education (it's a brilliant language for teaching programming!)
- Desktop GUIs (e.g., wxWidgets, Kivy, Qt)
- Software Development (build control, management, and testing – Scons, Buildbot, Apache Gump, Roundup, Trac)
- Business applications (ERP and e-commerce systems – Odoo, Tryton)
- Games (e.g., Battlefield series, Sid Meier's Civilization IV...), websites and services (e.g., Dropbox, UBER, Pinterest, BuzzFeed...)

Chapter 2

Artificial Intelligence

Mankind is welcoming the fourth industrial revolution represented by intelligent technology. New technologies such as AI integrated into all aspects of human society, driving change in global macro trends, such as sustainable social development and economic growth. New kinetic energy, smart city upgrading, industrial digital transformation, consumer experience, etc

2.1 About Artificial Intelligence

Artificial Intelligence (AI) is a new technical science that studies and develops theories, methods, techniques, and application systems for simulating and extending human intelligence. In 1956, the concept of AI was first proposed by John McCarthy, who defined the subject as "science and engineering of making intelligent machines, especially intelligent computer program". AI is concerned with making machines work in an intelligent way, similar to the way that the human mind works. At present, AI has become an interdisciplinary course that involves various fields.

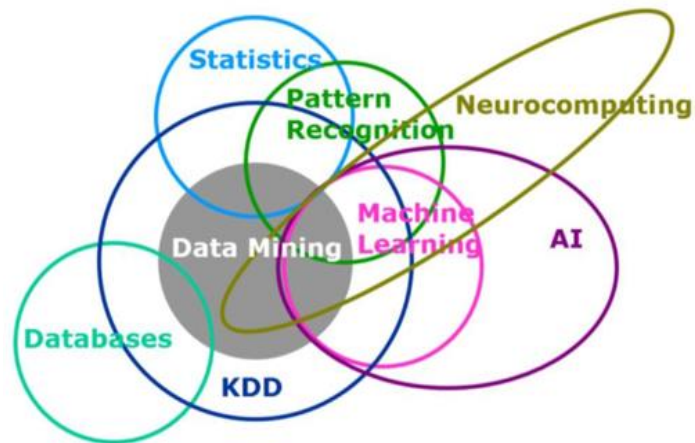
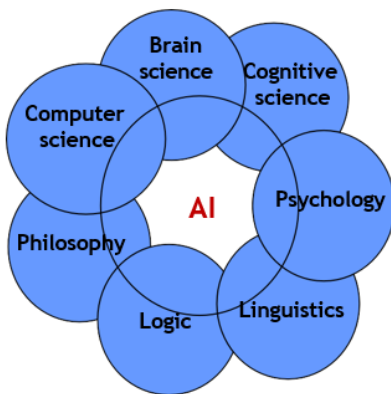


Fig 2.1: Application of AI

Fig 2.2: Concepts related to AI

2.2 History of Artificial Intelligence

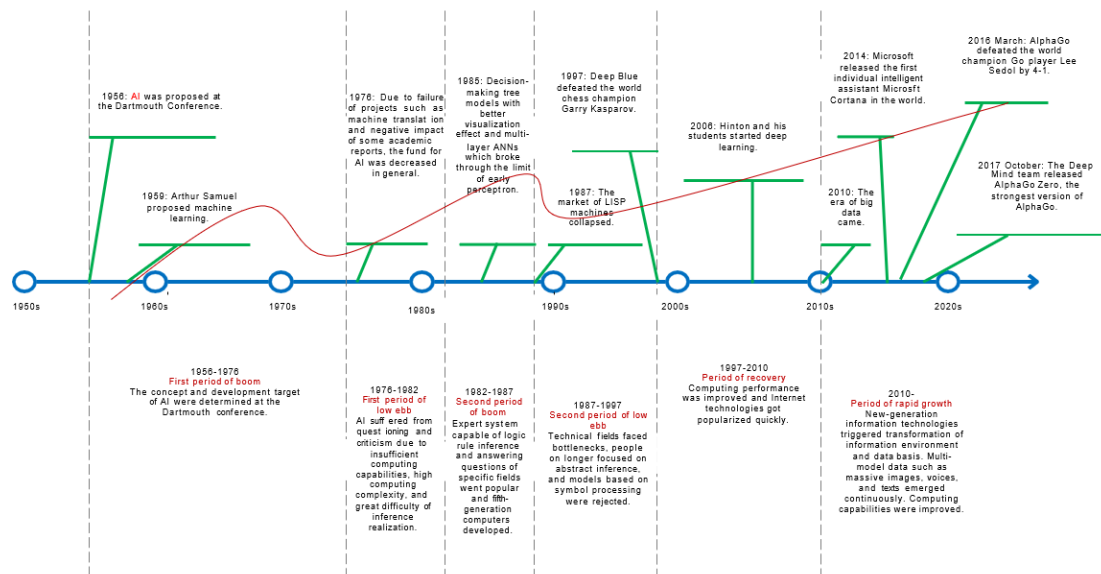


Fig 2.3: History of AI.

From 1957 to 1974, AI flourished. Computers could store more information and became faster, cheaper, and more accessible. Machine learning algorithms also improved and people got better at knowing which algorithm to apply to their problem. Early demonstrations such as Newell and Simon's General Problem Solver and Joseph Weizenbaum's ELIZA showed promise toward the goals of problem solving and the interpretation of spoken language respectively. These successes, as well as the advocacy of leading researchers (namely the attendees of the DSRPAI) convinced government agencies such as the Defense Advanced Research Projects Agency (DARPA) to fund AI research at several institutions. The government was particularly interested in a machine that could transcribe and translate spoken language as well as high throughput data processing. Optimism was high and expectations were even higher. In 1970 Marvin Minsky told Life Magazine, "from three to eight years we will have a machine with the general intelligence of an average human being." However, while the basic proof of principle was there, there was still a long way to go before the end goals of natural language processing, abstract thinking, and self-recognition could be achieved.

In the 1980's, AI was reignited by two sources: an expansion of the algorithmic toolkit, and a boost of funds. John Hopfield and David Rumelhart popularized "deep learning" techniques which allowed computers to learn using experience. On the other hand Edward Feigenbaum

introduced expert systems which mimicked the decision making process of a human expert. The program would ask an expert in a field how to respond in a given situation, and once this was learned for virtually every situation, non-experts could receive advice from that program. Expert systems were widely used in industries. The Japanese government heavily funded expert systems and other AI related endeavors as part of their Fifth Generation Computer Project (FGCP). From 1982-1990, they invested \$400 million dollars with the goals of revolutionizing computer processing, implementing logic programming, and improving artificial intelligence. Unfortunately, most of the ambitious goals were not met. However, it could be argued that the indirect effects of the FGCP inspired a talented young generation of engineers and scientists. Regardless, funding of the FGCP ceased, and AI fell out of the limelight.

Ironically, in the absence of government funding and public hype, AI thrived. During the 1990s and 2000s, many of the landmark goals of artificial intelligence had been achieved. In 1997, reigning world chess champion and grand master Gary Kasparov was defeated by IBM's Deep Blue, a chess playing computer program. This highly publicized match was the first time a reigning world chess champion loss to a computer and served as a huge step towards an artificially intelligent decision-making program. In the same year, speech recognition software, developed by Dragon Systems, was implemented on Windows. This was another great step forward but in the direction of the spoken language interpretation endeavor. It seemed that there wasn't a problem machines couldn't handle. Even human emotion was fair game as evidenced by Kismet, a robot developed by Cynthia Breazeal that could recognize and display emotions.

One could imagine interacting with an expert system in a fluid conversation, or having a conversation in two different languages being translated in real time. We can also expect to see driverless cars on the road in the next twenty years (and that is conservative). In the long term, the goal is general intelligence, that is a machine that surpasses human cognitive abilities in all tasks. This is along the lines of the sentient robot we are used to seeing in movies. Even if the capability is there, the ethical questions would serve as a strong barrier against fruition. When that time comes (but better even before the time comes), we will need to have a serious conversation about machine policy and ethics (ironically both fundamentally human subjects), but for now, we'll allow AI to steadily improve and run amok in society.

2.3 Types of Artificial Intelligence

Generally, AI can be classified as two distinct types. They are as:

- *Strong AI*: The strong AI view holds that it is possible to create intelligent machines that can really reason and solve problems. Such machines are considered to be conscious and self-aware, can independently think about problems and work out optimal solutions to problems, have their own system of values and world views, and have all the same instincts as living things, such as survival and security needs. It can be regarded as a new civilization in a certain sense.
- *Weak AI*: The weak AI view holds that intelligent machines cannot really reason and solve problems. These machines only look intelligent, but do not have real intelligence or self-awareness.

2.4 Applications of Artificial Intelligence

At present, application directions of AI technologies mainly include:

- *Computer vision*: a science of how to make computers "see".
- *Speech processing*: a general term for various processing technologies used to research the voicing process, statistical features of speech signals, speech recognition, machine-based speech synthesis, and speech perception.
- *Natural language processing (NLP)*: a subject that use computer technologies to understand and use natural language

Also, these technologies can be categorized as sub-fields according to their use in that field.



Fig 2.4: Application Fields of AI.

Chapter 3

Machine Learning

Machine learning is a core research field of AI, and it is also a necessary knowledge for deep learning.

3.1 About Machine Learning

Machine learning (including deep learning) is a study of learning algorithms. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .

3.2 Difference - Human Learning & Machine Learning

Humans acquire knowledge through experience either directly or shared by others. Machines acquire knowledge through experience shared in the form of past data.

Skill is a manifestation of intelligence possessed by humans. And intelligence is the ability to apply knowledge. Human intelligence sustains, but his knowledge fades as new technologies emerge. Humans without knowledge in particular subjects can apply their intelligence to solve problems in new domains. But machines can solve new problems only if their intelligence has been updated with retraining on data acquired from the changed scenarios. This is a fundamental difference between human intelligence and machine intelligence.

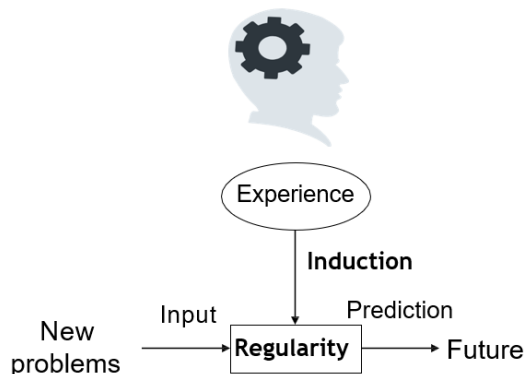


Fig 3.1: Human Learning

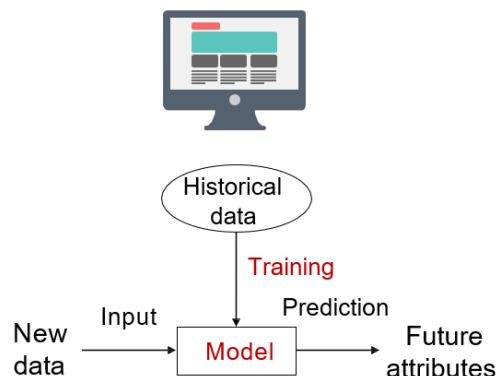


Fig 3.2: Machine Learning

3.3 Difference – Rule Based Approach & Machine Learning

In contrast with Rule-Based Approach Machine Learning is more advantageous in aspects of variable input values. For saying if about Rule Based approach, we have to manually define the set of rules on which the data is processed and output is produced. But on the other hand in case of Machine Learning we use data as for training our model & in accordance to data provided the model itself defines a boundary of rules that are complex but more effective as compared to Rule Based.

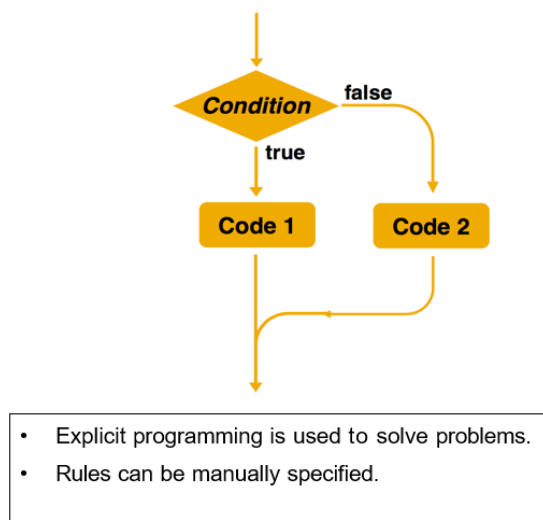


Fig 3.3: Rule Based Approach

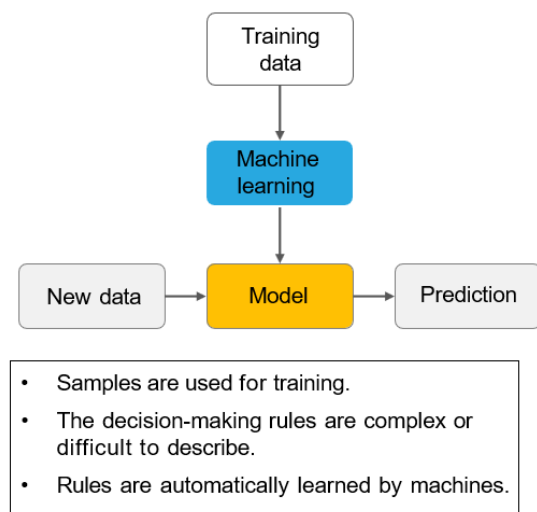


Fig 3.4: Machine Learning Approach

3.4 Problems Solved using Machine Learning

Machine learning can deal with many types of tasks. The following describes the most typical and common types of tasks.

- *Classification:*

A computer program needs to specify which of the k categories some input belongs to. To accomplish this task, learning algorithms usually output a function $f: R^n \rightarrow (1, 2, \dots, k)$. For example, the image classification algorithm in computer vision is developed to handle classification tasks.

- *Regression:*

For this type of task, a computer program predicts the output for the given input. Learning algorithms typically output a function $f: R^n \rightarrow R$. An example of this task type is to predict the claim amount of an insured person (to set the insurance premium) or predict the security price.

- *Clustering:*

A large amount of data from an unlabelled dataset is divided into multiple categories according to internal similarity of the data. Data in the same category is more similar than that in different categories. This feature can be used in scenarios such as image retrieval and user profile management.

3.5 Types of Machine Learning

In general, Machine Learning is classified in four types. They can be highlighted as:

- *Supervised learning:*

Obtain an optimal model with required performance through training and learning based on the samples of known categories. Then, use the model to map all inputs to outputs and check the output for the purpose of classifying unknown data.

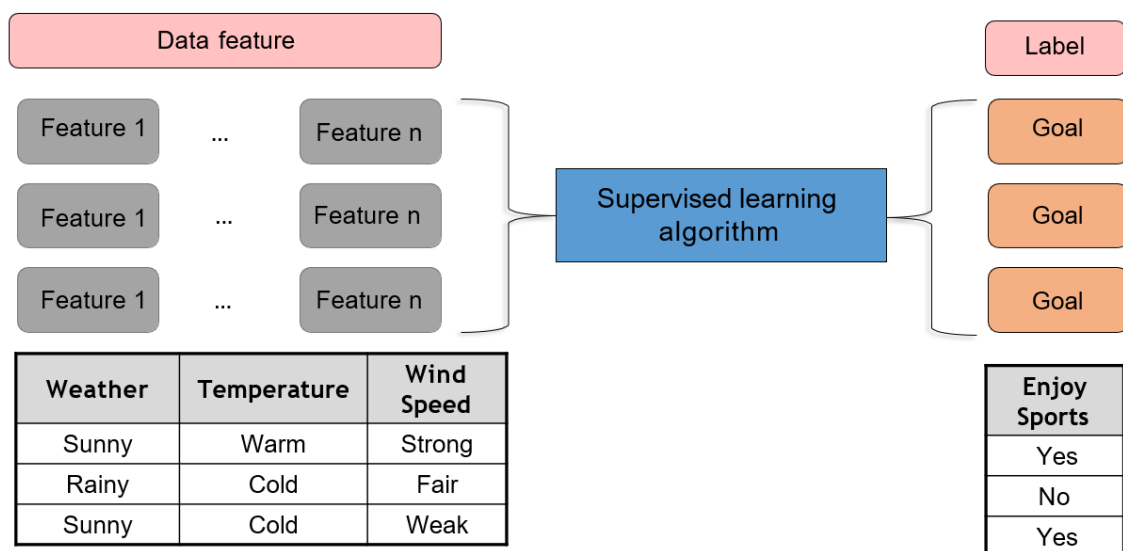


Fig 3.5: Supervised Machine Learning.

- *Unsupervised learning:*

For unlabeled samples, the learning algorithms directly model the input datasets. Clustering is a common form of unsupervised learning. We only need to put highly similar samples together, calculate the similarity between new samples and existing ones, and classify them by similarity.

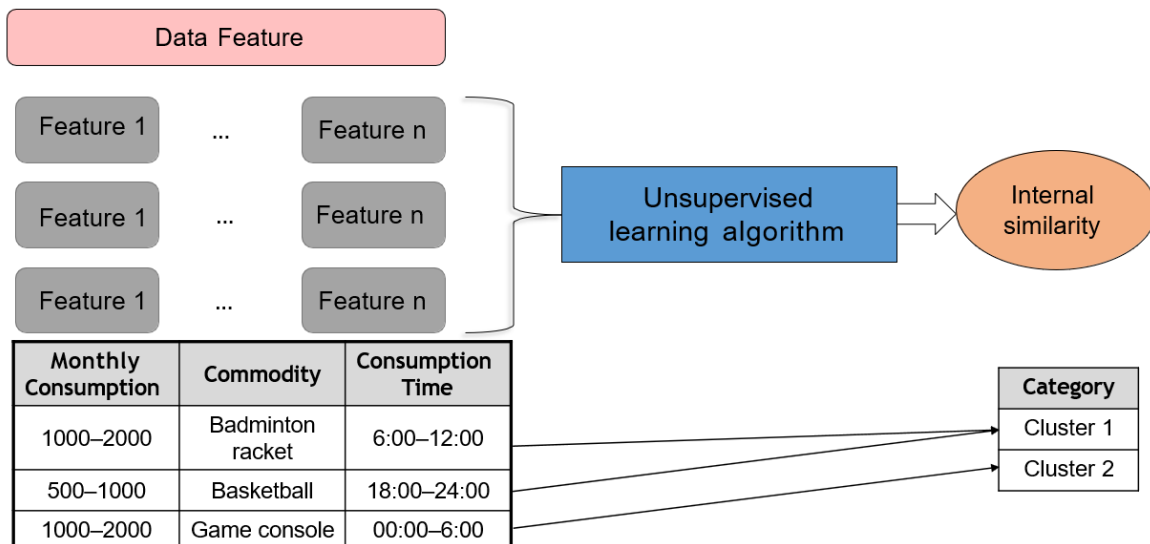


Fig 3.6: Unsupervised Machine Learning

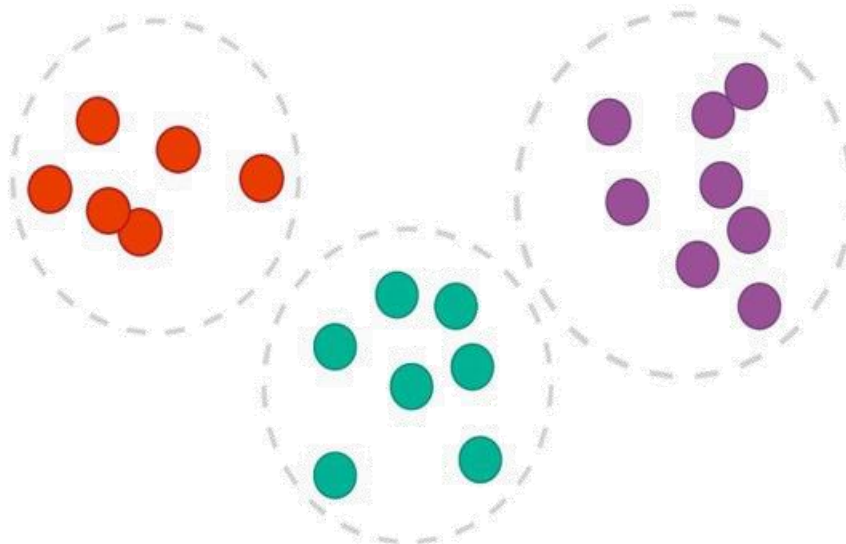


Fig 3.7: Unsupervised Machine Learning - Clustering

- *Semi-supervised learning:*

In one task, a machine learning model that automatically uses a large amount of unlabelled data to assist learning directly of a small amount of labelled data.

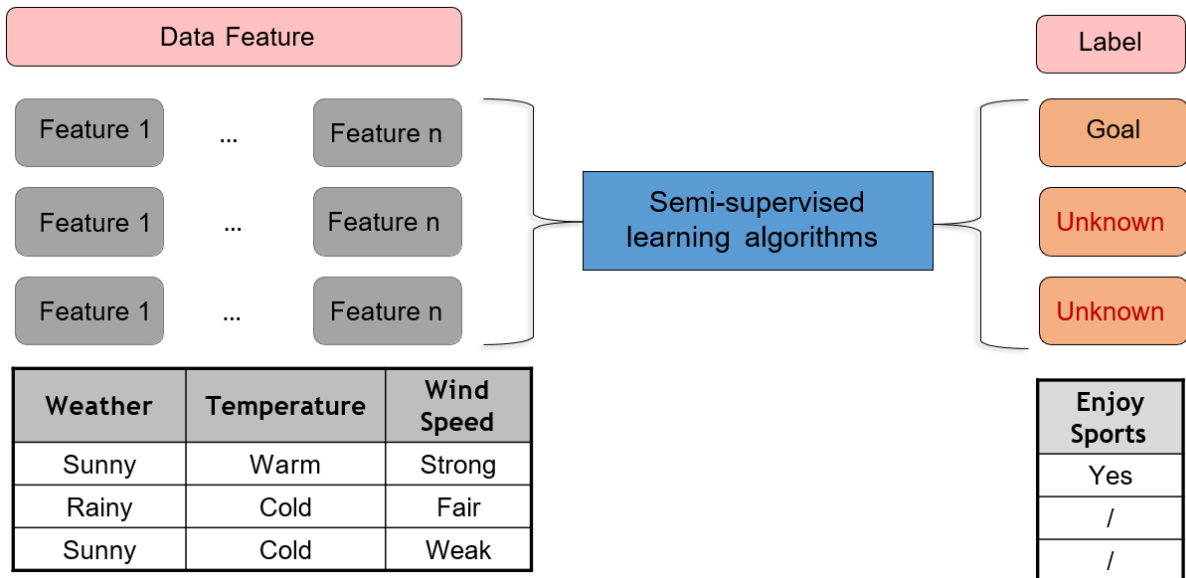


Fig 3.8: Semi-supervised Machine Learning.

- *Reinforcement learning:*

It is an area of machine learning concerned with how agents ought to take actions in an environment to maximize some notion of cumulative reward. The difference between reinforcement learning and supervised learning is the teacher signal. The reinforcement signal provided by the environment in reinforcement learning is used to evaluate the action (scalar signal) rather than telling the learning system how to perform correct actions.

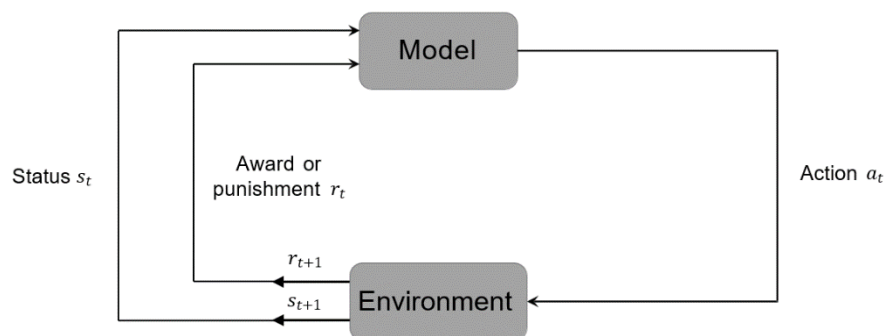


Fig 3.9 Reinforcement Learning.

3.6 Procedure of Machine Learning

The basic procedure of Model building through Machine Learning algorithm can be understood with the help of following flowchart:

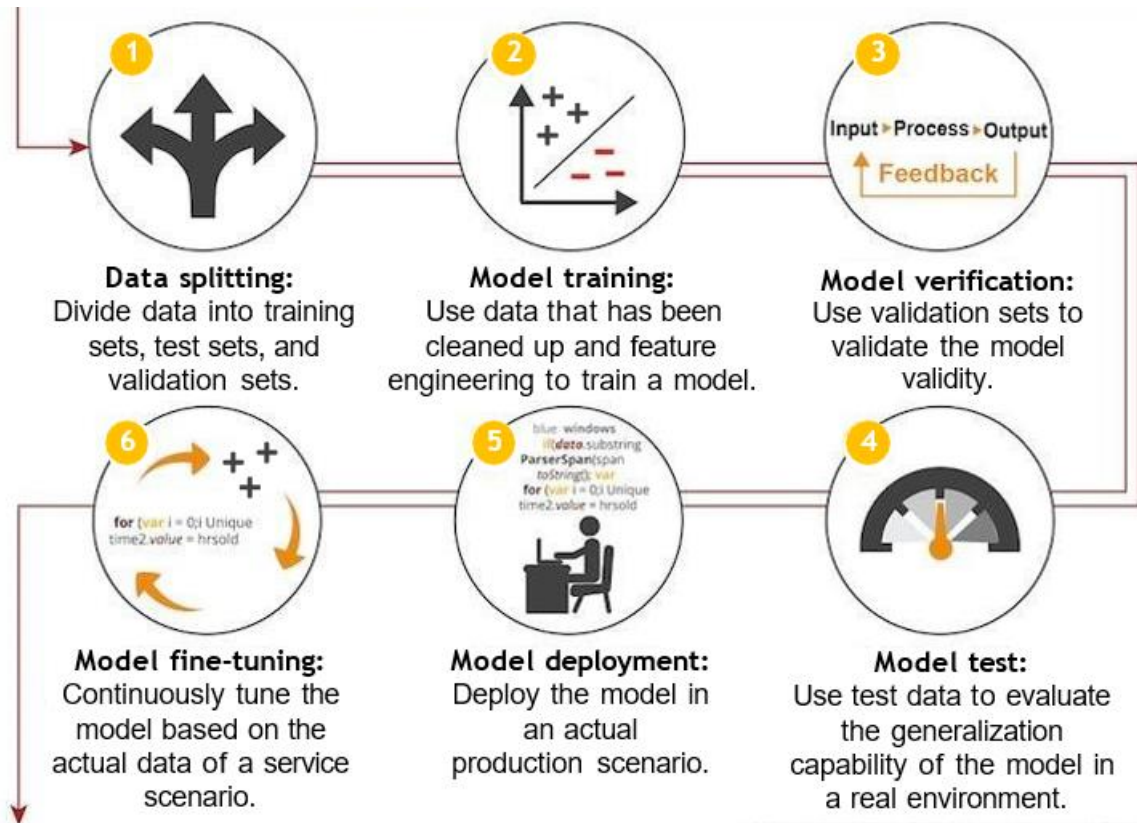


Fig 3.10: Machine Learning Process

Each mentioned step has its own significance for machine learning process and can affect the accuracy & efficiency of model if not configured correctly.

Chapter 4

Introduction To Sentiment Analysis of Incoming Calls on Help Desk

4.1 Understanding Sentiment Analysis:

- Definition: It's a technique that uses natural language processing (NLP) to automatically identify and analyze the emotional tone or sentiment expressed in text or speech.
- Goal: To uncover whether the expressed opinion is positive, negative, or neutral.

4.2 Applying Sentiment Analysis to Incoming Calls:

1. Call Recording: Help desks typically record calls for quality assurance and training purposes.
2. Speech-to-Text Transcription: Specialized software converts the audio of calls into text format.
3. Sentiment Analysis: NLP algorithms process the text to extract sentiment, often assigning scores or labels like "positive," "negative," or "neutral."

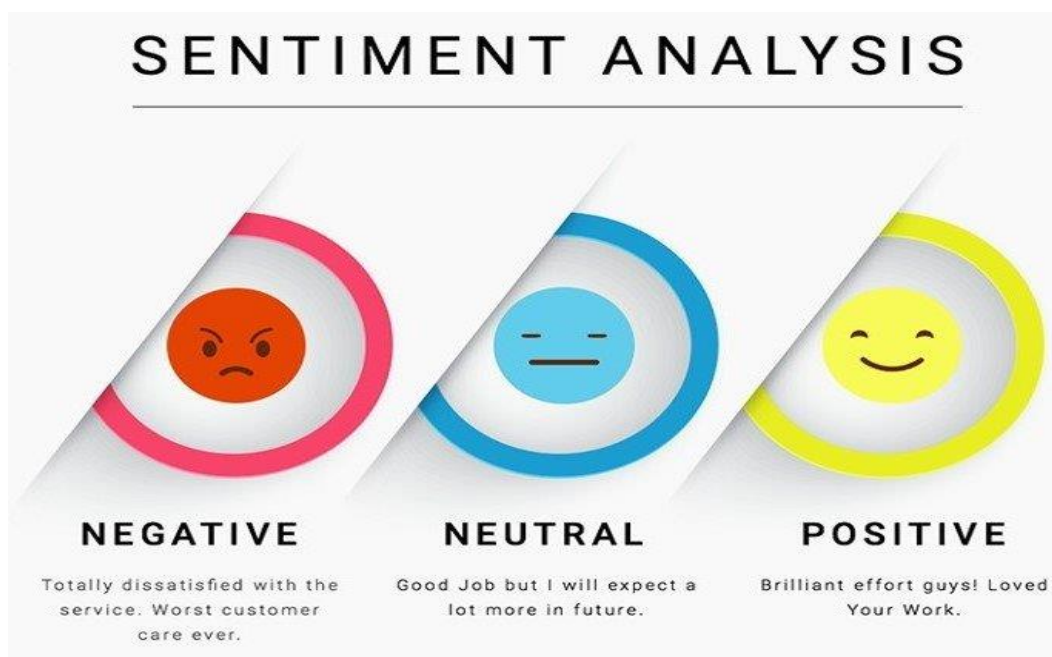


Fig 4.1: General Sentiments

4.3 Benefits for Help Desks:

- **Uncover Customer Sentiment:** Gain insights into overall customer satisfaction levels and identify specific pain points.
- **Improve Customer Service:** Tailor responses to address customer concerns effectively, leading to greater satisfaction.
- **Monitor Agent Performance:** Assess how agents handle calls and provide coaching for improvement.
- **Prioritize Urgent Calls:** Identify calls with high negative sentiment for immediate attention.
- **Personalize Customer Experiences:** Tailor responses based on sentiment to enhance satisfaction and build relationships.
- **Identify Trends and Patterns:** Track sentiment over time to uncover recurring issues, common frustrations, or areas for improvement.
- **Drive Business Decisions:** Use insights to guide product development, marketing strategies, and customer service initiatives.

4.4 Process Overview:

1. **Audio Recording and Transcription:** Calls are recorded and converted to text.
2. **Text Preprocessing:** Cleaning and preparing text for analysis (e.g., removing irrelevant words, correcting errors).
3. **Sentiment Analysis Using NLP Techniques:** Applying algorithms to classify sentiment.
4. **Visualizing Results:** Presenting sentiment data in charts and dashboards for easy understanding.
5. **Taking Action:** Using insights to improve customer service, agent performance, and overall business strategies.

4.5 Challenges and Considerations:

- **Accuracy of Speech-to-Text:** Errors in transcription can impact sentiment analysis results.
- **Nuances of Language:** Sarcasm, irony, and context can be difficult for algorithms

to detect accurately.

- Privacy Concerns: Handling sensitive customer data ethically and responsibly.
- Integration with Help Desk Systems: Ensuring seamless integration for real-time analysis and insights.

Sentiment analysis of incoming calls is a powerful tool for help desks to enhance customer experiences, improve operational efficiency, and drive business growth. By understanding customer sentiment, organizations can make informed decisions to address issues, improve service quality, and increase customer satisfaction.

Chapter 5

Implementation Highlights

Implementing sentiment analysis for incoming calls on a help desk can provide valuable insights into customer satisfaction and help improve the overall customer experience. Here are some implementation highlights for sentiment analysis in this context:

5.1 Data Collection

Data collection is a crucial and intricate phase in developing effective hate speech detection systems. The process involves gathering diverse and representative datasets that encapsulate the nuances and complexities of hate speech within various contexts. The challenges associated with collecting such data are multifaceted, ranging from defining hate speech to ensuring a balanced representation across demographics and cultural nuances.

To initiate the data collection process, it is essential to establish a clear and comprehensive definition of hate speech tailored to the specific context in which the system will operate. This definition serves as the foundation for annotating and labeling the datasets, guiding the selection of relevant examples that span a spectrum of offensive language, discriminatory content, and harmful expressions.

The identification of appropriate sources for data retrieval is a critical consideration. Social media platforms, online forums, and public communication channels serve as rich resources for capturing real-world instances of hate speech. However, careful attention must be paid to the ethical implications of utilizing user-generated content. Balancing the need for a diverse dataset with respect for user privacy and consent is paramount.

Annotation processes play a pivotal role in shaping the quality of the dataset. Skilled annotators, often well-versed in the cultural and contextual intricacies of the language, contribute to the nuanced labeling of hate speech instances. Annotation guidelines must be meticulously crafted, providing clarity on what constitutes hate speech, potential biases, and ambiguous cases.

Ensuring diversity in the dataset is essential for creating a robust hate speech detection model. This includes representing various demographics, linguistic styles, and cultural backgrounds

to avoid perpetuating biases in the system. Special attention should be given to underrepresented groups to prevent skewed outcomes.

The temporal dimension of data collection is also crucial. Hate speech evolves over time, adapting to societal changes and emerging trends. Therefore, datasets must be periodically updated to reflect the dynamic nature of online discourse.

In parallel, considerations for data imbalance and class distribution must be addressed. Hate speech instances may be less prevalent than non-hate speech, leading to imbalanced datasets. Techniques such as oversampling, under sampling, or employing advanced machine learning methodologies can help mitigate these challenges.

In summary, data collection for hate speech detection is a meticulous and ethical process that requires careful definition, diverse sourcing, thoughtful annotation, and ongoing updates. The quality and representativeness of the dataset directly influence the effectiveness and fairness of the ensuing hate speech detection model, emphasizing the need for a systematic and ethical approach throughout this critical phase of development.

5.2 Data Cleaning

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. It surely isn't the fanciest part of machine learning and at the same time, there aren't any hidden tricks or secrets to uncover. However, the success or failure of a project relies on proper data cleaning. Professional data scientists usually invest a very large portion of their time in this step because of the belief that "Better data beats fancier algorithms". If we have a well-cleaned dataset, there are chances that we can get achieve good results with simple algorithms also, which can prove very beneficial at times especially in terms of computation when the dataset size is large. Obviously, different types of data will require different types of cleaning. However, this systematic approach can always serve as a good starting point. The following are the most common steps involved in data cleaning:

1. Data inspection and exploration:

This step involves understanding the data by inspecting its structure, and identifying missing values, outliers, and inconsistencies.

2. Handling missing data:

Missing data is a common issue in real-world datasets, and it can occur due to various reasons such as human errors, system failures, or data collection issues. Various techniques can be used to handle missing data, such as imputation, deletion, or substitution.

3. Handling outliers:

Outliers are extreme values that deviate significantly from the majority of the data. They can negatively impact the analysis and model performance. Techniques such as clustering, interpolation, or transformation can be used to handle outliers.

4. Data transformation:

Data transformation involves converting the data from one form to another to make it more suitable for analysis. Techniques such as normalization, scaling, or encoding can be used to transform the data.

5. Data integration:

Data integration involves combining data from multiple sources into a single dataset to facilitate analysis. It involves handling inconsistencies, duplicates, and conflicts between the datasets.

5.2.1 Advantages of Data Cleaning

1. Improved model performance:

Data cleaning helps improve the performance of the ML model by removing errors, inconsistencies, and irrelevant data, which can help the model to better learn from the data.

2. Increased accuracy:

Data cleaning helps ensure that the data is accurate, consistent, and free of errors, which can help improve the accuracy of the ML model.

3. Better representation of the data:

Data cleaning allows the data to be transformed into a format that better represents the

underlying relationships and patterns in the data, making it easier for the ML model to learn from the data.

4. Improved data quality:

Data cleaning helps to improve the quality of the data, making it more reliable and accurate. This ensures that the machine learning models are trained on high-quality data, which can lead to better predictions and outcomes.

5. Improved data security:

Data cleaning can help to identify and remove sensitive or confidential information that could compromise data security. By eliminating this information, data cleaning can help to ensure that only the necessary and relevant data is used for machine learning.

5.3 Data Analysis and Exploration

Data analysis and exploration are integral components of developing effective hate speech detection systems. These processes involve delving into the intricacies of the collected datasets, understanding patterns, identifying trends, and uncovering insights to inform the creation of robust and accurate detection models.

5.3.1 Understanding the Data Landscape

The initial step in data analysis is to comprehend the overall landscape of the dataset. This includes examining the volume of data, the distribution of hate speech instances versus non-hate speech, and the prevalence of different linguistic styles and cultural nuances. Understanding the baseline characteristics sets the stage for subsequent exploration.

5.3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a critical phase that involves visualizing and summarizing key statistics to unveil underlying patterns. Visualization techniques, such as histograms, word clouds, and heatmaps, help in understanding the frequency and distribution of words and phrases within hate speech content. EDA aids in identifying potential biases, linguistic variations, and the contextual dynamics of hate speech instances.

5.3.3 Linguistic and Semantic Analysis

Analyzing the linguistic features of hate speech is crucial for model development. Natural Language Processing (NLP) techniques are employed to tokenize text, extract features, and identify linguistic patterns. Semantic analysis helps in understanding the context in which hate speech occurs, distinguishing between offensive language and non-hateful expressions that may share similar lexical structures.

5.3.4 Temporal Analysis

Hate speech is dynamic and may evolve over time. Temporal analysis involves examining the dataset across different time periods to identify trends, emerging topics, and shifts in language use. This temporal understanding is vital for creating models that can adapt to evolving patterns of hate speech.

5.3.5 User-Level Analysis

Analyzing hate speech at the user level provides insights into the behavior and patterns of individuals. This involves understanding the frequency of hate speech contributions by specific users, the relationships between users, and the potential influence of social networks on the propagation of hate speech.

5.3.6 Addressing Imbalances

Imbalances in the dataset, where hate speech instances may be less prevalent than non-hate speech, require careful attention. Techniques such as oversampling or undersampling can be employed to balance the dataset and avoid biases in model training.

5.3.7 Feature Engineering

Data analysis guides the process of feature engineering, where relevant features are selected or created to enhance the discriminatory power of the detection model. Features may include linguistic markers, sentiment analysis scores, and contextual information.

5.3.8 Preprocessing and Cleaning

Insights gained from data analysis inform preprocessing steps to clean the data. This involves handling missing values, removing irrelevant information, and addressing noise to enhance the quality of the dataset.

5.3.9 Validation and Model Iteration

The insights obtained from data analysis serve as a foundation for validating and fine-tuning the hate speech detection model. Continuous iteration is key, with adjustments made based on the performance observed during testing phases.

Data modeling involves creating a system that can recognize and filter out offensive language and discriminatory content online. This process is like teaching a computer to understand and identify harmful words or phrases. Imagine building a virtual brain for a computer. This brain needs to learn what hate speech looks like so it can spot it when people use it. We do this by feeding the computer lots of examples of both regular language and hate speech. It's like showing it good and bad examples to learn the difference.

To make this virtual brain smart, we use something called algorithms. These are like recipes for the computer, telling it step by step how to figure out if a sentence is okay or not. The computer looks at things like specific words, the way words are put together, and the overall meaning. Think of it as teaching a friend new words. You might say, "This word is bad, and this one is okay." The computer learns from these examples and becomes better at telling right from wrong. But it's not perfect from the start. Just like how you might need practice to get good at something, the computer needs lots of examples to become accurate. We give it more and more sentences, helping it understand the subtleties of language.

Once the computer has learned enough, it becomes a helpful tool. When you share something online, it checks if your words are respectful or if they might hurt someone. It's like having a digital friend reminding us to be kind and considerate in our words. However, it's important to know that this virtual brain isn't flawless. It might sometimes make mistakes, either missing some harmful words or flagging harmless ones. So, the computer keeps learning and getting better over time as it sees more examples.

In the end, data modeling for hate speech detection is like teaching a computer to be a good digital friend. It learns from examples, uses algorithms to understand language, and helps us create a more positive online environment.

Chapter 6

Modules & Libraries

6.1 Python Modules

A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.

6.2 Python Libraries

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

Some popular Python Libraries are as follows:

Here are key modules and libraries commonly used in sentiment analysis projects for incoming calls:

6.2.1 Audio Processing and Speech-to-Text:

- Libraries:
 - pyAudioAnalysis: Feature extraction, classification, segmentation of audio signals (Python)
 - Librosa: Audio analysis, feature extraction, manipulation (Python)
 - SpeechRecognition: Transcribe speech to text (Python)
 - Google Cloud Speech-to-Text API: Cloud-based speech recognition (various languages)
 - Amazon Transcribe: Cloud-based speech recognition (various languages)

6.2.2 Text Preprocessing:

- Libraries:
 - NLTK (Natural Language Toolkit): Tokenization, stemming, lemmatization, POS tagging (Python)
 - spaCy: Tokenization, named entity recognition, part-of-speech tagging, dependency parsing (Python)
 - TextBlob: Sentiment analysis, part-of-speech tagging, noun phrase extraction (Python)

6.2.3 Sentiment Analysis:

- Libraries:
 - TextBlob: Sentiment analysis, part-of-speech tagging, noun phrase extraction (Python)
 - VADER (Valence Aware Dictionary and sEntiment Reasoner): Lexical based sentiment analysis, tuned for social media text (Python)
 - NLTK: VADER is included in NLTK
 - Hugging Face Transformers: Pre-trained sentiment analysis models (e.g., BERT, RoBERTa) (Python)
 - Flair: NLP library with pre-trained sentiment analysis models (Python)

6.2.4 Additional Libraries:

- pandas: Data manipulation and analysis (Python)
- NumPy: Numerical computing (Python)
- scikit-learn: Machine learning algorithms (Python)
- TensorFlow/Keras: Deep learning frameworks (Python)
- PyTorch: Deep learning framework (Python)

Choosing the right modules and libraries depends on:

- Programming language: Python is a popular choice for sentiment analysis.
- Specific tasks: Audio processing, text preprocessing, sentiment analysis,

visualization.

- Accuracy requirements: Consider pre-trained models for higher accuracy.
- Deployment needs: Cloud-based services offer scalability and ease of deployment.
- Resources and expertise: Some libraries require more expertise and computational resources.

Recommendations:

- Start with NLTK and TextBlob: These are user-friendly libraries for basic sentiment analysis.
- Explore Hugging Face Transformers: For advanced sentiment analysis with pre-trained models.
- Consider cloud-based services: For ease of use and scalability.
- Experiment with different libraries: Find the best fit for your project's specific needs.

Chapter 7

Project Description

7.1 Problem Statement

The main purpose of the project is to return the emotion of a voice recording passed through it.

7.2 Workflow of Project

- Input audio file/recording.
- Normalization of input file.
- Passing of file through trained model.
- Label determination based on output of model.
- Printing of output.

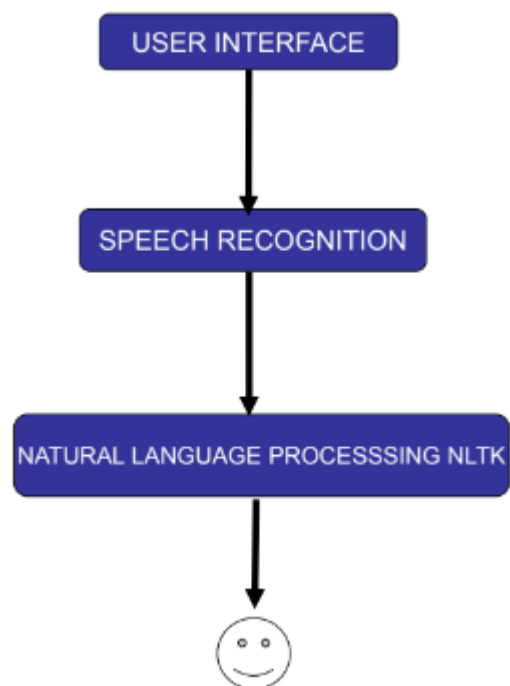


Fig 7.1: Workflow

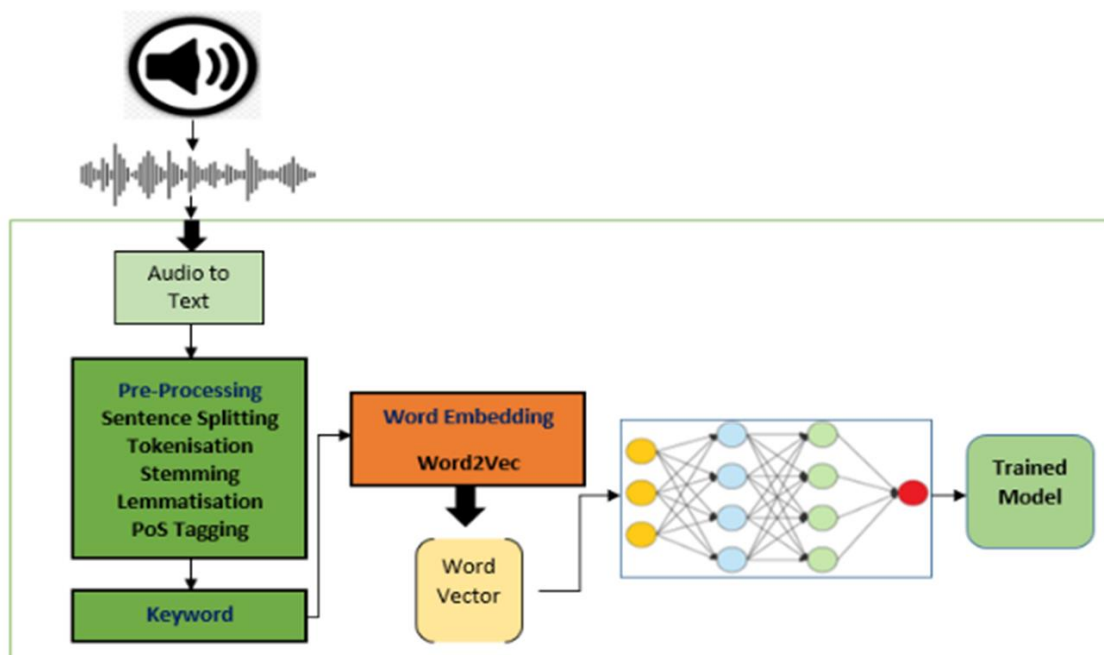


Fig 7.2: workflow model

7.3 Significance of important code segments.

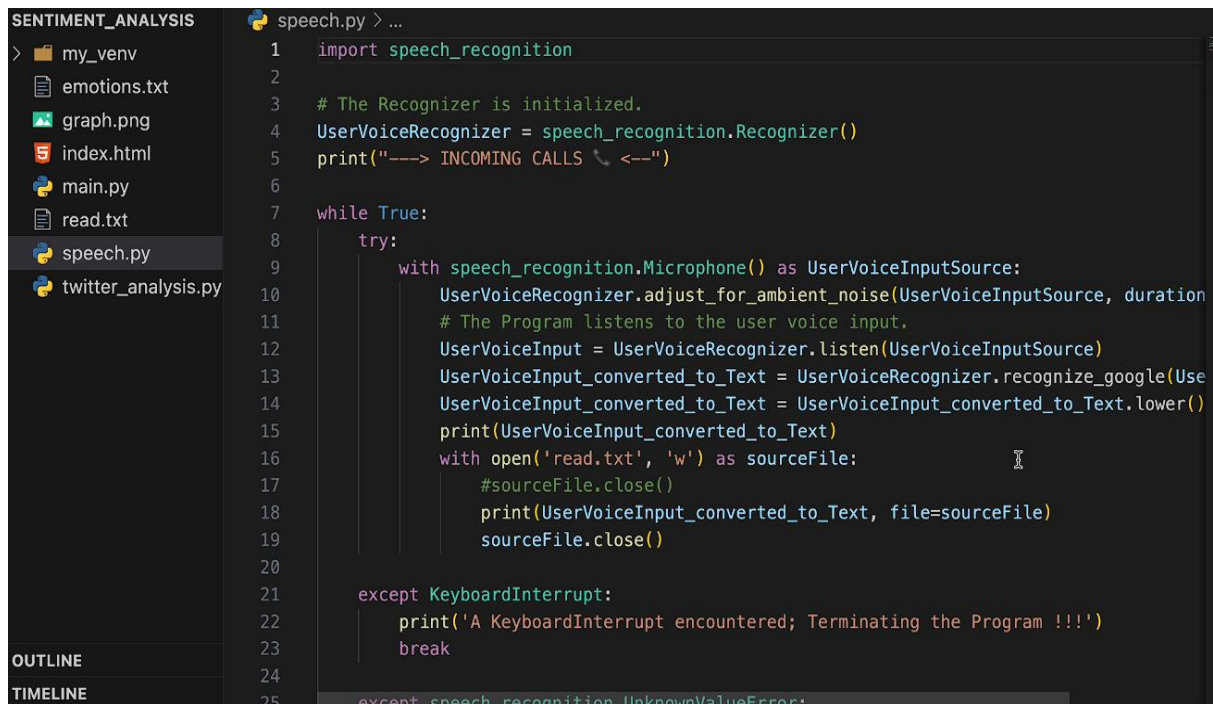
- *Model Definition.*

Fig 7.3: Model Definition

```

main.py > ...
1  import string
2  from collections import Counter
3
4  import matplotlib.pyplot as plt
5  from nltk.corpus import stopwords
6  from nltk.sentiment.vader import SentimentIntensityAnalyzer
7  from nltk.stem import WordNetLemmatizer
8  from nltk.tokenize import word_tokenize
9
10 text = open('read.txt', encoding='utf-8').read()
11 lower_case = text.lower()
12 cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))
13
14 # Using word_tokenize because it's faster than split()
15 tokenized_words = word_tokenize(cleaned_text, "english")
16
17 # Removing Stop Words
18 final_words = []
19 for word in tokenized_words:
20     if word not in stopwords.words('english'):
21         final_words.append(word)
22
23 # Lemmatization - From plural to single + Base form of a word (example better-> good)
24 lemma_words = []
  
```

- Speech to text converting



```

1  import speech_recognition
2
3  # The Recognizer is initialized.
4  UserVoiceRecognizer = speech_recognition.Recognizer()
5  print("----> INCOMING CALLS 📞 <---")
6
7  while True:
8      try:
9          with speech_recognition.Microphone() as UserVoiceInputSource:
10             UserVoiceRecognizer.adjust_for_ambient_noise(UserVoiceInputSource, duration=2)
11             # The Program listens to the user voice input.
12             UserVoiceInput = UserVoiceRecognizer.listen(UserVoiceInputSource)
13             UserVoiceInput_converted_to_Text = UserVoiceRecognizer.recognize_google(UserVoiceInput)
14             UserVoiceInput_converted_to_Text = UserVoiceInput_converted_to_Text.lower()
15             print(UserVoiceInput_converted_to_Text)
16             with open('read.txt', 'w') as sourceFile:
17                 #sourceFile.close()
18                 print(UserVoiceInput_converted_to_Text, file=sourceFile)
19                 sourceFile.close()
20
21             except KeyboardInterrupt:
22                 print('A KeyboardInterrupt encountered; Terminating the Program !!!')
23                 break
24
25             except speech_recognition.UnknownValueError:

```

Fig 7.4: generating a text file

- Graph of sentiment

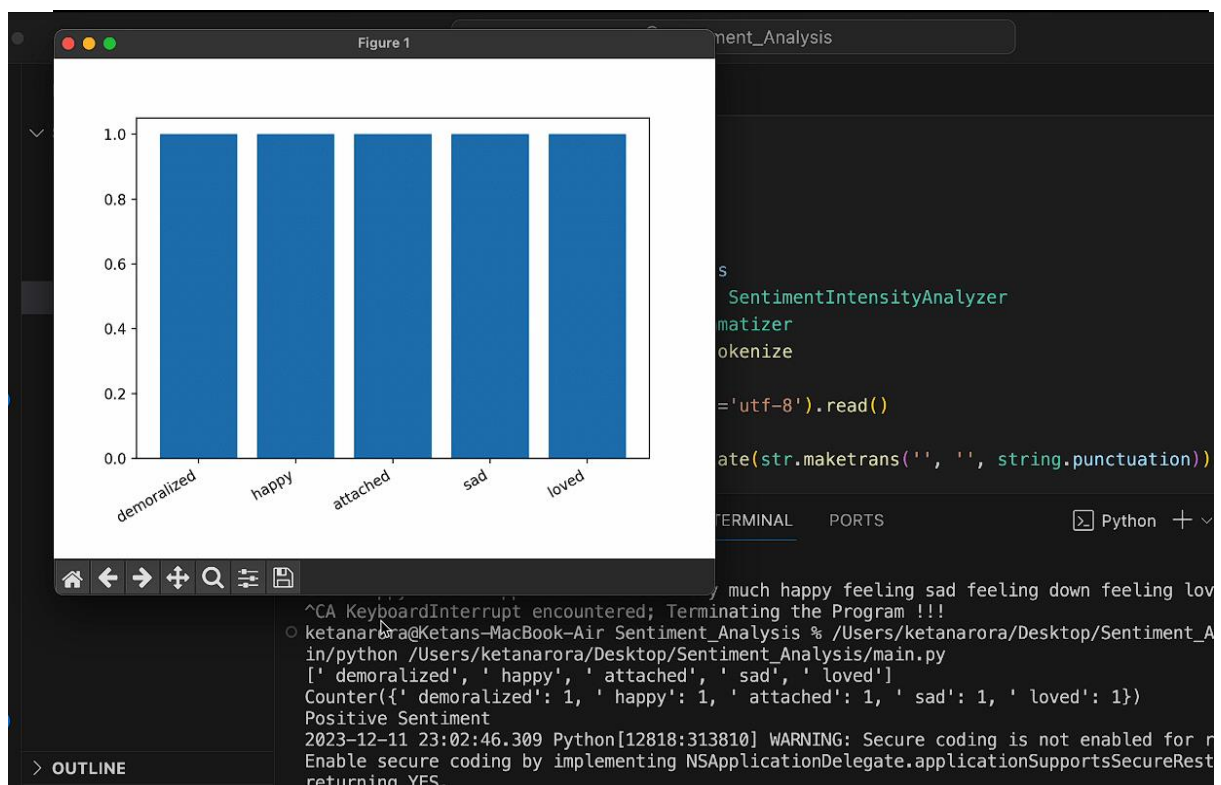


Fig 7.5: sentiment analysis graph

Conclusion

This project has shed light on the power of sentiment analysis in understanding and enhancing the customer experience within the helpdesk environment. By analyzing the emotional undertones of incoming calls, we have gained valuable insights beyond mere call outcomes, uncovering a treasure trove of information about customer satisfaction, key pain points, and areas for improvement.

1. Unveiling Customer Sentiment: The analysis revealed the overall sentiment towards the helpdesk, providing a crucial benchmark for understanding customer satisfaction. Additionally, we delved deeper, examining sentiment variations based on call reasons, highlighting specific areas of concern (e.g., technical issues) or areas of commendation (e.g., efficient resolutions).

2. Identifying Opportunities for Improvement: By analyzing the language used by customers, we identified both positive and negative keywords. Positive keywords like "helpful" or "efficient" offer valuable validation of our efforts, while negative keywords like "frustrated" or "unresolved" point towards areas where we can strengthen our support.

3. Agent Performance Insights: The analysis went beyond the macro level to shed light on individual agent performance. We identified agents who consistently receive positive feedback, allowing us to recognize their strengths and share best practices. Conversely, we also identified areas where coaching and support can be provided to improve agent-customer interactions.

4. Data-Driven Recommendations: Based on these insights, the report provides actionable recommendations for improving customer satisfaction, agent training, and overall helpdesk operations. These recommendations, grounded in data-driven insights, pave the way for tangible improvements in customer experience.

5. Continuous Improvement: The report emphasizes the importance of ongoing sentiment analysis as a critical tool for monitoring progress and identifying emerging issues. By continuously gathering and analyzing customer feedback, we can maintain a pulse on their experience and adapt our approach to consistently meet their evolving needs.

In conclusion, this project has demonstrated the immense value of incorporating sentiment analysis into the helpdesk ecosystem. By understanding and responding to the emotional tone of customer interactions, we can build a more customer-centric support system, foster stronger relationships, and ultimately drive long-term business success.

Remember to customize this conclusion further by incorporating specific findings and recommendations from your project. You can also tailor the language to match your audience and desired tone.

References

- Python: Guido van Rossum and the Python development team. (2021). Python Language Reference, version 3.10.9. Retrieved from <https://docs.python.org/3/reference/index.html>
- Librosa: Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. (2021). librosa: Audio and Music Signal Analysis in Python. Journal of Open Source Software, 6(1), p.24. doi: 10.21105/joss.02493
- TensorFlow: Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. (2021). TensorFlow: An end-to-end open source machine learning platform. Retrieved from <https://www.tensorflow.org/>
- TESS: The TESS (The Emotional Speech Set) dataset is available on Kaggle at the following link: <https://www.kaggle.com/ejlok1/tess-the-emotional-speech-set>
- RAVDESS: The RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset is available on Kaggle at the following link: <https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio>
- SAVEE: The SAVEE (Surrey Audio-Visual Expressed Emotion) dataset is available on Kaggle at the following link: <https://www.kaggle.com/ejlok1/savee-surrey-audio-visual-expressed-emotion>
- Research Article: [Voice acoustic measures of depression severity and treatment response collected via interactive voice response \(IVR\) technology.](#)