AI BASED CUSTOMER INTENT CLASSIFICATION

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENTS FOR THE AWARD OF THE DEGREE

MASTER OF SCIENCE IN DATA ANALYTICS

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

HAREESH C S

220011023643

Department of Mathematics & Statistics

BASELIUS COLLEGE KOTTAYAM

(NAAC Re-accredited with A++ Grade)

Kottayam, Kerala – 686 001

2024

A Project Report on

AI BASED
CUSTOMER INTENT CLASSIFICATION

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF SCIENCE IN DATA ANALYTICS

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

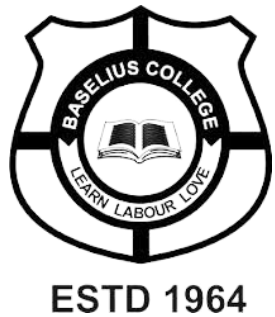
HAREESH C S

220011023643

Under the guidance and supervision of

Ms. Ross Cyriac

Lecturer
Department of Mathematics & Statistics
Baselius College Kottayam



ESTD 1964

BASELIUS COLLEGE KOTTAYAM

(NAAC Re-accredited with A++ Grade)

Kottayam Kerala – 686 001

2024

BASELIUS COLLEGE KOTTAYAM

(NAAC Re-accredited with A++ Grade)

Kottayam, Kerala – 686 001

ESTD 1964

Department of Mathematics & Statistics

CERTIFICATE

This is to certify that the project work entitled **"AI BASED CUSTOMER INTENT CLASSIFICATION"** is a bonafide record of work done by **HAREESH C S, 220011023643,** in partial fulfilment of the requirements for the award of Degree of **MASTER OF SCIENCE IN DATA ANALYTICS** during the academic year 2023-2024.

Ms. Ross Cyriac
Lecturer
Dept of Mathematics & Statistics
Baselius College Kottayam

Ms. Preetha Mathew
Head Of The Department
Dept of Mathematics & Statistics
Baselius College Kottayam

External Examiner
Place: Kottayam
Date:

DECLARATION

I, **HAREESH C S (Reg No: 220011023643)** hereby declare that this project work entitled **"AI BASED CUSTOMER INTENT CLASSIFICATION"** is a record of original work done by me under the guidance of **Ms. Ross Cyriac**, Lecturer, Department of Mathematics and Statistics and the work has not formed the basis for the award of any degree or diploma or similar title to any candidate of any university subject.

Place: Kottayam **HAREESH C S**

Date:

ACKNOWLEDGEMENT

This project is not complete if one fails to acknowledge all who have been instrumental in the successful completion of the project. If words were to be the symbol of undiluted feelings and token of gratitude then let the words play the heralding role of expressing my gratitude.

First of all, I thank the **"God Almighty"** for his immense grace and blessings in my life and at each stage of this project.

I express my sincere and profound gratitude to **Prof. Dr. Biju Thomas**, Principal, Baselius college Kottayam, for providing all the facilities during the period of the project.

I extend my gratitude to **Ms. Preetha Mathew**, Head of the Department, Department of Mathematics and Statistics, who is a constant source of inspiration and whose advice helped me to complete this project successfully.

I express my deep sense of gratitude to my internal project guide **Ms. Ross Cyriac**, Lecturer, Department of Mathematics and Statistics, for her profound guidance for the successful completion of this project.

With great enthusiasm I express my gratitude to all the faculty members of the Department of Mathematics and Statistics for their timely help and support.

I would like to thank **SMEClabs, Kaloor**, for providing me with an invaluable learning experience during my project in Data Science. It has been a pleasure to work with the staff, and I have gained much practical knowledge about Data Science.

Finally, I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project successfully.

HAREESH C S

ABSTRACT

The goal of this project is to create a smart computer system that can understand and categorize customer messages accurately. This will help businesses provide better and more personalized customer service, leading to increased customer satisfaction and loyalty.

The project uses a large dataset of customer queries and their intents (like complaints or inquiries) to train a computer model. This model relies on advanced technologies such as Natural Language Processing (NLP) and Deep Learning Artificial Neural Network (ANN) to analyze and understand customer messages.

To achieve this, the team uses tools like the Natural Language Toolkit (NLTK) for text preprocessing and TensorFlow for deep learning-based text classification. The project is implemented using Python version 3.9.0 or above, and popular coding environments like Jupyter Notebook, Google Colab, and Visual Studio Code are used.

Once the model is developed, it will be integrated into the customer service process to make responses more efficient. The team plans to continually improve the model by gathering feedback and providing comprehensive documentation and training for customer service teams. The project aims to enhance overall customer support systems and improve the overall customer experience for businesses.

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

In a Customer Intent Classification project, the goal is to understand and categorize the intentions behind customer queries or messages. This helps businesses effectively respond to customer needs and streamline their services. By using techniques like Artificial Neural Networks (ANN), we create a smart system capable of automatically figuring out the purpose or intent behind each customer interaction.

In customer service, businesses are using artificial intelligence (AI) more to improve how they help customers. One important part of this is using Customer Intent Classification systems. Create a smart system that understands what customers are saying and sorts it into categories. This helps businesses analyze customer feedback and improve their products or services. This project aims to enhance customer service, automate responses, and ultimately improve the overall customer experience.

1.1 OVERVIEW OF THE PROJECT

The project will focus on the development of a Customer Intent Classification model using Natural Language Processing (NLP) techniques. The data set is from Kaggle. The Customer Support Intent Dataset is a large-scale collection of customer support queries and their corresponding intents. It contains thousands of examples of customer queries, such as complaints, inquiries, and requests, along with their respective intent categories, such as billing, technical support, and feedback. The dataset is particularly useful for developing and training natural language processing (NLP) models that can accurately classify customer support queries into their corresponding intents. This dataset can be used by businesses and organizations to improve their customer support systems, optimize their operations, and provide better customer experiences.

Here, we're making different types of models using different methods like K Nearest Neighbors (KNN), Decision Tree, Random Forest, and Artificial Neural Network (ANN). We pick the one with the best accuracy, then test it with a sample prediction.

We made a model with Keras using Artificial Neural Network (ANN). It is a computational model inspired by the structure and functioning of biological neural networks in the human brain. Artificial Neural Networks are a key component of machine learning and deep learning, used for tasks such as pattern recognition, classification, regression, and decision-making. This typically

involves an input layer, one or more hidden layers, and an output layer. The first layer gets the input data, and each part of this layer shows a specific detail about the input. In the middle, there can be hidden layers. Neurons in these layers handle the input using connections with weights and special functions to make things a bit more complex. Each part of a hidden layer shows something the system learned from the input. The last layer gives the final result of the network. Choose appropriate activation functions, such as ReLU for hidden layers and softmax for the output layer in the case of multi-class classification.

1.2 RELEVANCE OF THE PROJECT

Understanding what customers want is crucial for businesses to succeed in today's competitive world. When businesses know exactly what their customers need, they can respond quickly and accurately, making customers happier and more loyal. This leads to long-term success because happy customers are likely to come back and recommend the business to others.

Knowing what customers want also helps sales teams use their resources better. By focusing on the most promising leads, sales processes become smoother and more efficient, leading to more sales and fewer wasted resources.

Listening to customer inquiries and feedback gives valuable insights into what's popular, what customers like, and what they don't. With this information, businesses can make products that match what customers want, making them happier and more competitive in the market.

Using technology to understand what customers want saves time and ensures decisions are consistent and accurate. By analyzing lots of data, businesses can find patterns and trends they might miss otherwise. Businesses that are good at understanding and meeting customer needs stand out from their competitors by providing great experiences. This helps them get more customers and become stronger in their industry.

Finally, analyzing customer data helps businesses make smarter decisions across different areas like marketing and product development. This makes sure resources are used well and that plans match what customers want, leading to better results and growth.

2. SYSTEM STUDY

2. SYSTEM STUDY

2.1. PROPOSED SYSTEM

First, we start by getting our data from Kaggle, a famous website for datasets. Then, we need to prepare our data before we can analyze it. This involves cleaning it up by removing any weird symbols or common words that don't add much meaning. We also change the words into numbers so our computer can understand them better. This makes it easier for us to study and understand the data later on.

In our project, we're trying to understand what customers like. We use different computer techniques to help us do this. These techniques include K-Nearest Neighbors, Random Forest, Decision Tree, and Artificial Neural Network. Each of these techniques helps us find patterns and learn from the data.

The dataset we're using has 6539 rows and 4 columns. It's carefully put together to include important details about what customers like. This data is really important because it's the foundation of our machine learning models. Also, the Customer Intent Classification datasets we use are already split into parts for training and testing. This makes it easier for us to develop our models. By using these datasets, our machine learning models are good at predicting what customers want. This helps us make products and services that suit them better.

2.2. CHALLENGES IN MODEL BUILDING

Customers frequently utilize language that is ambiguous or imprecise, rendering their intentions unclear. Moreover, the interpretation of the same words can fluctuate significantly depending on the context, leading to discrepancies across different situations. User queries exhibit a wide spectrum of variation and often lack a standardized structure, thereby posing considerable challenges in addressing misspellings, slang, abbreviations, or colloquial language.

Within customer interactions, sensitive information is commonly exchanged, underscoring the critical importance of prioritizing the privacy and security of customer data when leveraging it for training AI models. With the escalating volume of customer interactions, scalable solutions become indispensable for managing the burgeoning data load and ensuring the delivery of real-time or near-real-time responses.

2.3. LIMITATION OF THE SYSTEM

Our project encounters significant challenges due to the dataset sourced from Kaggle. Despite its substantial size, boasting 6539 rows and 4 columns, uncertainties persist regarding its completeness and accuracy. This lack of certainty poses a considerable risk to the accuracy and usefulness of our models. Furthermore, relying exclusively on data from online shopping platforms raises concerns about the representation of individuals who refrain from sharing their opinions online. This potential bias could skew our findings and compromise the reliability of our analyses, undermining the credibility of our project.

Moreover, dealing with complicated models such as Artificial Neural Networks (ANN) poses significant challenges that are difficult to overcome. Insufficient access to computational resources, time constraints, and a shortage of personnel proficient in model improvement exacerbate these challenges. However, openly addressing these issues in our project documentation is paramount. This transparency enables us to grasp the capabilities and limitations of our system, facilitating the development of strategies for improvement. Recognizing its limitations allows us to find ways to make it work better, leading to significant improvements in our project results.

3. SYSTEM DESIGN

3. SYSTEM DESIGN

3.1. ABOUT THE LANGUAGE

The analysis and model building are done using Python 3.9.0. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum in 1991. Python's main focus is on making code easy to read. It's good for making programs quickly and for connecting different parts of programs together. Python has simple rules for writing code, which makes it easier to maintain programs. It also supports using modules and packages, which helps keep programs organized and allows for reusing code. Python is free to use and can be installed on any major computer platform.

3.2. ABOUT THE EDITOR

Google Colab, or Google Colaboratory, is a tool provided by Google that lets people write and run Python code in a web browser. It's really popular among data scientists and people who work with machine learning because it's free and gives access to special computers called GPUs and TPUs, which make training machine learning models faster. With Colab, you can write and run Python code in a special kind of document called a Jupyter Notebook. These notebooks are great for doing things like analyzing data or exploring ideas because you can mix together text, code, and pictures. Colab also works well with Google Drive, so you can easily save and share your work. It comes with lots of useful Python tools already installed, like NumPy, pandas, TensorFlow, and scikit-learn. Plus, it lets you use powerful hardware to speed up your work, especially when training complex machine learning models. And if you're working with others, Colab lets you collaborate in real-time, making it easy to work together on projects.

3.3. DATASET FOR THE STUDY

The data set is from Kaggle. The Customer Support Intent Dataset is a big bunch of customer questions and what they're really asking for. The dataset contains information in 4 columns and 6539 rows. The attributes in the data set are utterance, intent, category, and tags. It's got lots of different types of questions, like complaints or questions about billing or technical issues. This dataset helps make computer programs understand what customers want when they ask questions. It's great for training programs to sort out different types of customer questions. Businesses can use this data to make their customer service better, fix problems faster, and make customers happier.

3.4. ABOUT NLP

Natural language processing (NLP) is like teaching computers to understand human language. It's a mix of computer science and artificial intelligence. NLP studies how language works and uses different methods like statistics and machine learning to help computers understand text or speech. With NLP, computers can figure out what people mean when they talk or write, including their feelings or intentions.

NLP is behind lots of useful stuff we use every day, like translating text, recognizing speech, summarizing text, and chatting with bots. You might have used some of these, like GPS systems that talk to you, digital assistants like Siri or Alexa, or software that types what you say. Businesses also use NLP to do things faster and better by tackling tasks involving language.

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful.

3.5. ABOUT ANN

Artificial Neural Networks are made up of artificial neurons called units. These units are grouped together in layers to form the network. A layer can have just a few units or many, depending on how complex the network needs to be to understand the data. Typically, a neural network has an input layer, a hidden layer or layers, and an output layer. The input layer gets data from the outside world for the network to analyze. This data goes through the hidden layers, which transform it into a form that's useful for the output layer. Finally, the output layer gives the network's response to the input data.

In most neural networks, units in one layer are connected to units in the next layer. These connections have weights that show how much one unit affects another. As data moves through the units, the neural network learns more about it, leading to an output from the output layer. Artificial neural networks function much like our brains, employing layers of interconnected nodes to process and analyze information. Commonly referred to as neural networks or neural nets, these structures consist of various layers, commencing with the input layer that receives external data. Subsequently, the information progresses to the hidden layer, where each neuron

integrates inputs from the previous layer, adjusting their significance through assigned weights before transmitting them onward. These weights undergo continual adjustments during training to optimize the network's performance. Through the intricate arrangement of layers and weight adjustments, neural networks demonstrate remarkable versatility, capable of tasks ranging from pattern recognition to complex decision-making. Their ability to learn from data enables continuous improvement, enhancing their efficacy over time. Neural networks find application across diverse domains, including image and speech recognition, owing to their adaptability and learning capabilities. As advancements in neural network architecture and training methodologies evolve, the potential for these systems to revolutionize industries and spur innovation remains substantial. Envisioning a future where intelligent machines collaborate harmoniously with humans to address significant challenges and pioneer new frontiers underscores the transformative potential of neural networks.

4. ALGORITHMS USED

4. ALGORITHMS USED

Artificial intelligence (AI) is the concept of creating machines that can perform tasks that typically require human intelligence. Machine learning is a type of artificial intelligence (AI) that enables computers to learn and improve from experience without being explicitly programmed. Deep learning is a subset of machine learning, which is itself a branch of artificial intelligence (AI). It involves training artificial neural networks with large amounts of data to recognize patterns and make predictions, similar to the way the human brain works. In the Customer Intent Classification Dataset, we use both deep learning and machine learning methods.

Machine learning is like teaching computers to learn from examples and get better at tasks over time, without being told exactly what to do. Instead of following specific instructions like in traditional programming, machine learning algorithms figure out patterns in data and use that to make predictions. It's like learning from experience.

Deep learning is a type of machine learning that's really good at tasks like recognizing images, understanding speech, and processing language. Deep learning models can automatically learn important features from the data, which makes them really useful for these kinds of tasks. Some common types of deep learning models are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

The main algorithms used in this project are :

- K Nearest Neighbors (KNN)
- Random Forest Classifier
- Decision Tree Classifier
- Artificial Neural Network (ANN)

K Nearest Neighbors (KNN)

The k-nearest neighbors (KNN) algorithm is a widely used method in machine learning for classification and regression tasks. It finds the k closest examples in the dataset based on some distance measure like Euclidean or Manhattan distance. For classification, it assigns the most common class label among the k neighbors to the new data point, while for regression, it computes the average value of the k neighbors. The choice of parameter k affects the algorithm's performance, with smaller values leading to more complex decision boundaries and potential overfitting, and larger values resulting in smoother boundaries but possibly overlooking local patterns. The distance metric used to measure similarity between data points, such as Euclidean,

Manhattan, or cosine similarity, is also important and depends on the nature of the data.

Random Forest Classifier

The Random Forest classifier is a smart computer program that helps make predictions. It does this by creating lots of small decision trees, each based on different parts of the data it learns from. These trees work together to give a final guess for each new thing it looks at. When we make these trees, we can decide how many we want and set some rules for how they should be made.

Random Forest works like a big team of trees. Each tree makes its own guess, and then all the guesses are put together to decide on the final answer. This teamwork idea is called the "wisdom of crowds," where having lots of different opinions can help make better decisions. One cool thing about Random Forest is that it can make many different trees, each looking at the data in its own way. This helps it understand different parts of the data better and make smarter guesses. Because Random Forest is so good at making accurate predictions, it's often chosen over just using one decision tree in many different computer programs. Its flexibility and reliability make it a favorite in all kinds of applications.

Decision Tree Classifier

A decision tree classifier is a handy tool in the world of teaching computers how to make choices based on examples they've seen. Picture it like a tree with branches, where each branch is a choice made based on rules or conditions. These choices help the classifier figure out what category or group a piece of information belongs to.

People like using decision trees in machine learning because they're simple and easy to understand. They can handle different kinds of information, like categories (like colors) or numbers (like ages). They're even smart enough to work with missing data. But here's the thing: decision trees can sometimes focus too much on small details. This can make them not so good at predicting new, unseen information. To fix this, we've got a few tricks. We can trim down the tree, getting rid of unnecessary branches to keep it simpler. We can also stop the tree from going too deep, or we can set rules about how much information each branch needs before making a choice. And if none of that works, we can team up with other decision trees in a group called random forests. Together, they're better at making predictions. In simple terms, decision trees are like the helpful guide in the world of machine learning. They're versatile, adaptable, and can tackle any challenge with some adjustments and collaboration.

Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are like digital brain cells that team up to handle information. These networks are made of artificial neurons, called units, grouped into layers. Depending on how tough the task is, a layer might have just a few neurons or a bunch.

There are usually three layers in a neural network: the input layer, the hidden layer(s), and the output layer. The input layer takes in data from the world that the network needs to understand. This data moves through the hidden layers, which process it to help make sense of it all. Then, the output layer gives the network's answer or guess based on the input. In most neural networks, neurons in one layer connect to neurons in the next layer. These connections have weights, which decide how much one neuron affects another. As data moves through, the network learns more, leading to a final answer from the output layer.

In the training of Artificial Neural Networks (ANNs), several important steps help make the network better. First, at the start, we set the weights and biases between neurons to random values to get things going. Then, we give the network some data, and it starts making guesses about what the outcome might be. After that, we check how good these guesses are by comparing them to the real answers using a special formula called a loss function. If the guesses aren't great, we tweak the weights and biases using a process called backpropagation. This helps the network learn from its mistakes and get better at making predictions. We do this over and over again until the network gets really good at making predictions. Once it's trained, the network can confidently predict outcomes for new data it hasn't seen before. This way, ANNs become reliable tools for solving real-world problems.

5. ANALYSIS AND INTERPRETATION

5. ANALYSIS AND INTERPRETATION

5.1. PURPOSE AND WORKFLOW

The process involves different steps, starting with handling the data, then moving to visualizing it, and next, analyzing and categorizing it. Exploratory data analysis (EDA) helps us understand the data better by summarizing its main features, often using graphs and charts. This gives us insights beyond just the formal analysis or testing of ideas. In this study, we're trying out various classification methods on the data to see how well they work. The steps include loading the dataset, looking at visualizations and analyzing the data, cleaning and preparing it, and finally, building classification models. First, we analyze reviews and draw some conclusions based on them. We also examine different ways of visualizing the data. Then, we try building a classification model using various algorithms and pick the one with the best accuracy.

5.2. PYTHON IN DATASCIENCE PROJECT

Python is like a versatile tool that can be used in many different ways for coding. It supports different styles of programming, like making objects, organizing code, and working with functions. Python is known for being easy to read and understand, so it's not too hard to learn. There are also lots of special libraries made just for analyzing data, which means people who work with data in different fields can find tools that are already made for what they need. Because Python can do so many things and is used by lots of people, it was only a matter of time before someone started using it for analyzing data.

Modules and Packages

In this project, we use many pre-made tools called modules and packages to help us analyze and visualize the data. Some of the modules we use are:

- **Numpy :** It will provide the support for efficient numerical computation.
- **Pandas :** It is convenient library that supports data-frames. Working with pandas will bring ease in many crucial data operations.
- **Matplotlib :** It provides a MATLAB-like plotting framework.
- **Seaborn :** It is a visualization library based on matplotlib which provides a high-level interface for drawing attractive statistical graphics.
- **Sklearn :** python library for data mining, data analysis and machine learning.

- **Wordcloud :** Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.
- **nltk :** The Natural Language Toolkit (NLTK) is a Python package for natural language processing.
- **TensorFlow :** It is an open-source software framework developed by Google Brain which allows developers to build and train machine learning models by creating computational graphs, which represent the operations that the model performs.
- **Keras :** It is an open-source deep learning library written in Python. It is used for implementing neural networks.

5.3. DATA CLEANING AND PREPROCESSING

First, we loaded the dataset using a function called read_csv from a library called pandas. This function helps us bring in data from a web link and turn it into a special kind of table called a dataframe. Then, we used another function called head() to see the first bit of the dataset and get an idea of what it looks like. We also used some other functions from pandas to check how big the dataset is and learn more about it. There are total 6539 observations with 4 different feature variables/attributes present in the customer intent classification dataset. After that, we looked at the data to see if there were any missing values using a function called isnull() in the pandas library. We found that there weren't any missing values in any of the parts of the data. It seems like the data was gathered carefully, or someone already cleaned it up before sharing it.

During preprocessing the text is first converted to lowercase. Then, anything in square brackets is removed using a regular expression pattern match. All punctuation marks are removed using another regular expression pattern match. Words containing digits are removed using yet another regular expression pattern match. Certain characters such as hyphens, brackets, and quotes are removed using another regular expression pattern match. Finally, newline characters are removed from the text.

5.4. DATA VISUALIZATIONS AND INTERPRETATIONS

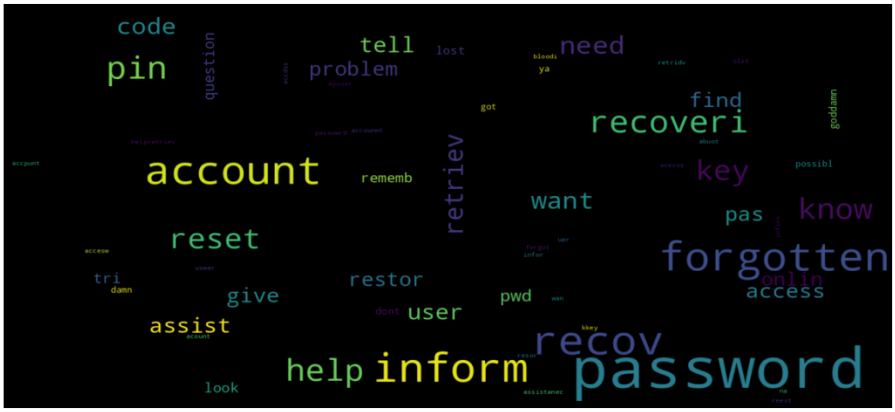
Data visualization means showing data in pictures to see patterns and connections that might not be obvious just by looking at the numbers. Python has lots of good libraries that help make different kinds of graphs. Whether you want a graph you can play with, a moving graph, or one that looks exactly how you want, Python has the tools.

In this project, we used a few of these libraries to make graphs that helped us understand our data better.

- **Matplotlib :** Gives you a lot of control over how your graphs look.
- **Seaborn :** Easier to use and comes with nice styles already built-in.

WordCloud

A word cloud is a visual representation of text data where the size of each word indicates its frequency or importance within the dataset. Commonly, words that appear more frequently in the text are displayed larger and more prominently. Word clouds are often used to quickly identify the most prominent terms in a body of text, making them useful for visualizing key themes or concepts.

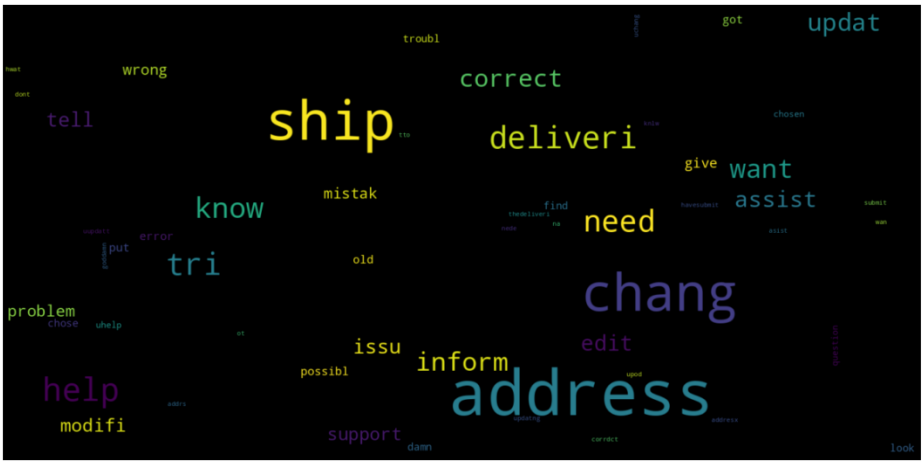


recover_password

Library : matplotlib, wordcloud

Plot used : wordcloud

Result : Most frequently occurring word in recover_password are password, recover, inform, forgotten, account, etc. Here customer's intention is to recover forgotten password of customer's account. That's why these words are most frequently seen here.



change_shipping_address

Library : matplotlib, wordcloud

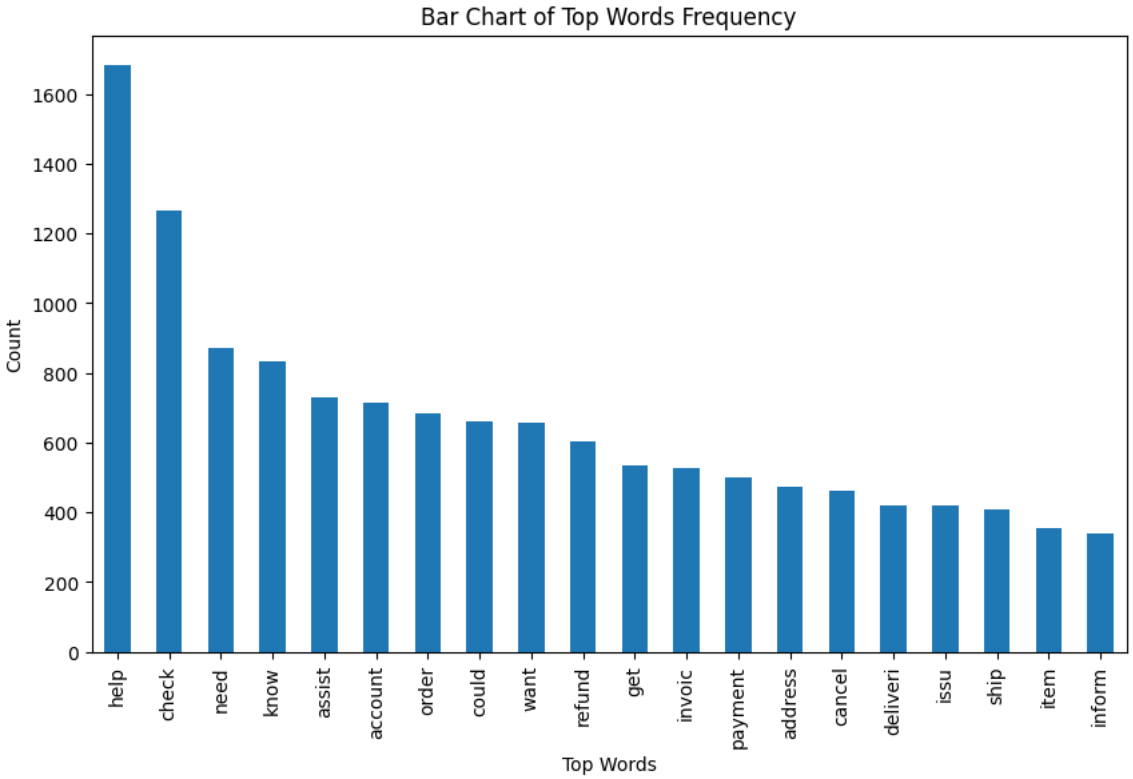
Plot used : wordcloud

Result : Most frequently occurring word in change_shipping_address are ship, address, chang, deliveri, correct, want, need, etc. Here the customer's need is to change the shipping address of the item to be delivered. That's why these words are most frequently seen here.

Bar Plot

A bar plot, also called a bar chart, is a way to show data using bars of different heights. Each bar represents a category, and the taller the bar, the bigger the number it represents. Usually, the bars go across the bottom of the graph, and the height of each one shows how much of something there is.

Bar plots are super helpful because they let you compare things easily. You can see which category has the most or the least just by looking at the bars. They're used in lots of different areas, like math, science, and business, to help people understand information quickly. Whether you're looking at how many apples are sold in different stores or comparing the heights of different buildings, bar plots make it easy to see what's going on with the data.



Library : matplotlib

Plot used : barplot

Result : The bar plot above shows the most common words in the reviews. This helps us understand what people are talking about and their opinions. We can see that most frequently used word is help. Most people needed was help. Hence, the word help is the most used word here. Other commonly used words in the utterance is check, need, and know.

6. MODEL BUILDING

6. MODEL BUILDING

6.1. USE OF ALGORITHMS IN MODEL BUILDING

In finance, it's really important to understand why models decide what they do. That's why scientists often choose simpler machine learning models, even if they're not super accurate. But sometimes, more complex models like ensembles and neural networks are better at predicting things, even though they're harder to understand. When we want to figure out how customers feel from their reviews, we use a tool called natural language processing in Python to build these models.

6.2. USE OF NLP IN MODEL BUILDING

To handle natural language, we use a tool called NLTK (Natural Language Toolkit) in Python.

Stop words

A stop word is a commonly used word (such as “the”, “a”, “an”, or “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Eliminating stop words can improve the accuracy and relevance of NLP tasks by drawing attention to the more important words, or content words.

Corpus

A corpus is a collection of text documents stored in directories, sometimes accompanied by other directories. When there are multiple collections, they are called 'corpora.' They come from things like digital text, audio transcripts, or scanned documents. Corpora are super important for studying how we use language every day. They help us understand how we communicate by showing the small details and patterns in our language.

Snowball Stemmer

The Snowball Stemmer can handle words from different languages, not just English. It works fast and can handle small pieces of text well. It's more powerful than the Porter Stemmer and is sometimes called the Porter2 Stemmer. Because it's better than the Porter Stemmer, the Snowball Stemmer works faster.

6.3. DATA PROCESSING AND SPLITTING

The customer intent classification dataset was previously split into train and test datasets. To find its split ratio, first, it computes the total number of samples by summing the row counts of both

the training and testing datasets. This total samples variable ensures an accurate representation of the entire dataset. Next, the split ratio is calculated by dividing the number of samples in the training dataset by the total samples. This ratio delineates the proportion of samples assigned to the training subset, thereby determining the distribution between training and testing data. Finally, a formatted print statement conveys the split ratio in percentage format, showcasing the percentage of samples allocated to each subset. 80% of it was training data and 20% testing data.

6.4. BUILDING MODELS

In Python, there are different methods for analyzing and predicting in data science. In this project, we mainly use these algorithms:

- K Nearest Neighbors (KNN)
- Random Forest Classifier
- Decision Tree Classifier
- ANN (Artificial Neural Network)

To decide which model to use, we compare their accuracy scores. Accuracy is a simple way to measure how well a model predicts outcomes correctly compared to all the outcomes. Precision is how many of the predicted positive outcomes are actually positive. Recall is how many of the actual positive outcomes the model predicted correctly.

K Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) stands out as a favored method for machine learning tasks, renowned for its simplicity and versatility in handling various data types without extensive rule-building. The essence of KNN lies in its intuitive approach: leveraging the proximity of data points to make informed predictions. The parameter "K" in KNN determines the number of neighboring points considered when predicting the class or value of a new data point.

When confronted with new data, KNN swiftly identifies the nearest known data points, drawing upon their characteristics to formulate a prediction. For classification tasks, it discerns the predominant group among these proximate points and assigns the new data point accordingly. Conversely, in regression scenarios, KNN computes the average of the nearby known values to estimate the target value for the new data point. The classification report is as follows:

	precision	recall	f1-score	support
class 0	0.86	0.96	0.91	25
class 1	0.70	0.72	0.71	39
class 2	0.85	0.80	0.83	41
class 3	0.75	0.81	0.78	26
class 4	0.80	0.90	0.85	31
class 5	0.92	0.92	0.92	26
class 6	0.75	0.97	0.85	37
class 7	0.62	0.65	0.64	23
class 8	0.70	0.67	0.68	24
class 9	0.91	0.74	0.82	42
class 10	0.74	0.56	0.64	25
class 11	0.88	0.79	0.84	29
class 12	0.88	0.88	0.88	32
class 13	0.89	0.89	0.89	38
class 14	0.89	0.69	0.77	35
class 15	0.82	0.90	0.86	31
class 16	0.90	0.96	0.93	28
class 17	0.80	0.70	0.74	23
class 18	0.82	1.00	0.90	27
class 19	0.94	0.97	0.95	30
class 20	0.62	0.80	0.70	20
class 21	0.68	0.56	0.61	27
class 22	0.94	0.97	0.95	32
class 23	0.89	0.97	0.93	32
class 24	0.84	0.72	0.78	36
class 25	0.88	0.71	0.79	31
class 26	0.81	0.89	0.85	28
accuracy			0.82	818
macro avg	0.82	0.82	0.81	818
weighted avg	0.82	0.82	0.82	818

Random Forest Classifier

The Random Forest Classifier is a type of machine learning algorithm used for classification tasks. It's called "random forest" because it's made up of a large number of decision trees that work together to make predictions. Each decision tree is trained on a random subset of the data, and then they vote to decide on the final prediction. This ensemble approach tends to produce more accurate and robust predictions compared to individual decision trees. The classification report is as follows:

	precision	recall	f1-score	support
class 0	0.93	1.00	0.96	25
class 1	0.86	0.92	0.89	39
class 2	0.98	0.98	0.98	41
class 3	0.82	0.88	0.85	26
class 4	0.91	0.97	0.94	31
class 5	0.89	0.92	0.91	26
class 6	0.95	0.97	0.96	37
class 7	0.74	0.74	0.74	23
class 8	0.88	0.96	0.92	24
class 9	0.92	0.81	0.86	42
class 10	0.63	0.76	0.69	25
class 11	0.82	0.79	0.81	29
class 12	0.94	1.00	0.97	32
class 13	0.97	0.97	0.97	38
class 14	0.94	0.86	0.90	35
class 15	0.97	0.97	0.97	31
class 16	1.00	0.96	0.98	28
class 17	0.89	0.74	0.81	23
class 18	0.96	1.00	0.98	27
class 19	0.88	1.00	0.94	30
class 20	0.81	0.85	0.83	20
class 21	0.96	0.81	0.88	27
class 22	0.97	1.00	0.98	32
class 23	1.00	1.00	1.00	32
class 24	0.93	0.75	0.83	36
class 25	1.00	0.94	0.97	31
class 26	1.00	1.00	1.00	28
accuracy			0.91	818
macro avg	0.91	0.91	0.91	818
weighted avg	0.92	0.91	0.91	818

Decision Tree Classifier

The Decision Tree Classifier is like a flowchart that helps computers make decisions. It's called a "decision tree" because it looks like a tree, with branches leading to different choices. Each choice is based on a feature of the data, like age or color. At the end of each branch, there's a final decision, or outcome, like "yes" or "no."

When the algorithm needs to make a prediction, it starts at the top of the tree and follows the branches that match the data until it reaches a final decision. Decision trees are easy to understand, which makes them helpful for explaining why a computer made a particular prediction. They can handle different types of data, like numbers or categories, which makes them useful for many kinds of problems. However, decision trees can sometimes remember the training data too well, which can cause problems when trying to predict new data. Techniques like pruning (trimming the tree) and using groups of trees together, like in Random Forests, can help fix this issue. The classification report is as follows:

	precision	recall	f1-score	support
class 0	1.00	1.00	1.00	25
class 1	0.97	1.00	0.99	39
class 2	0.98	0.98	0.98	41
class 3	0.96	0.96	0.96	26
class 4	1.00	1.00	1.00	31
class 5	0.96	1.00	0.98	26
class 6	1.00	1.00	1.00	37
class 7	0.95	0.87	0.91	23
class 8	0.92	1.00	0.96	24
class 9	1.00	0.95	0.98	42
class 10	0.86	0.72	0.78	25
class 11	0.86	0.83	0.84	29
class 12	1.00	1.00	1.00	32
class 13	1.00	0.97	0.99	38
class 14	0.95	1.00	0.97	35
class 15	1.00	1.00	1.00	31
class 16	1.00	0.96	0.98	28
class 17	1.00	0.96	0.98	23
class 18	1.00	1.00	1.00	27
class 19	1.00	1.00	1.00	30
class 20	0.86	0.95	0.90	20
class 21	0.90	0.96	0.93	27
class 22	0.97	1.00	0.98	32
class 23	1.00	1.00	1.00	32
class 24	0.94	0.94	0.94	36
class 25	1.00	1.00	1.00	31
class 26	0.97	1.00	0.98	28
accuracy			0.97	818
macro avg	0.96	0.97	0.96	818
weighted avg	0.97	0.97	0.97	818

Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) works like our brain to handle tough jobs like recognizing patterns and sorting things into groups. It's made up of layers of connected nodes called neurons. Each neuron takes in information, processes it, and passes it on to the next layer. The connections between neurons have weights that affect the output. By training, the network adjusts these weights to get better at its tasks. ANNs are great at learning complex patterns from

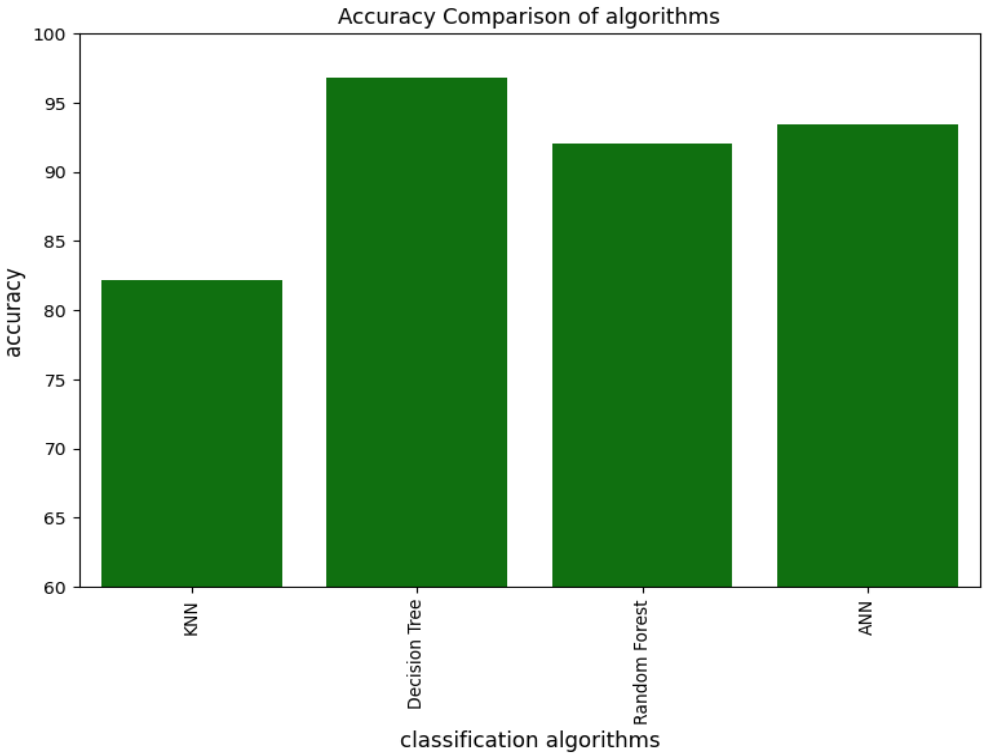
data, which helps in tasks like recognizing images and understanding language. But training them needs a lot of computer power and data, and they're harder to understand compared to simpler models like decision trees.

```
26/26 [==============================] - 0s 2ms/step - loss: 0.5366 - accuracy: 0.9340
Test loss: 0.53655606508255
Test accuracy: 0.9339853525161743
```

6.5. COMPARISON OF ALGORITHMS

From the above model building the accuracy score is compared as follows.

- K Nearest Neighbors (KNN) - 82.15 %
- Decision Tree - 96.89 %
- Random Forest - 92.05 %
- ANN - 93.40 %



7. FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENT

Adding self-learning abilities to AI models is a big step forward in technology. It means these models can keep getting better over time by learning from how people interact with them. This helps them understand what users want and respond better. So, the AI can update itself without needing constant human input, making it more useful in different situations.

One important way to use self-learning AI is in handling job applications. Here, AI can help speed up the hiring process by sorting through resumes and messages from applicants. It figures out what each person wants and directs them to the right department or person. This makes things faster and ensures that each applicant gets the help they need quickly.

Another helpful idea is to include humans in the AI process. This means that when the AI faces tricky situations it doesn't understand, people can step in and give their input. This helps the AI learn and make better decisions over time. So, by working together, AI and humans can keep improving how AI systems work.

Overall, combining self-learning AI, automated job application handling, and human feedback is a powerful way to make AI better in many areas. It speeds up tasks, makes them more efficient, and helps AI adapt to different situations. As technology advances, using these methods will be important for making even more progress in AI.

8. CONCLUSION

8. CONCLUSION

In this project, our aim was to develop a robust system capable of discerning customers' needs from their messages. Through rigorous testing, we gained profound insights into the language patterns and requirements of our clientele. Our endeavor illustrates how leveraging computational techniques can decode the underlying intent behind customer messages, thereby empowering companies to enhance their service delivery, operational efficiency, and ultimately, foster business expansion.

Our enthusiasm for this system stems from its potential to revolutionize customer interactions and streamline business processes. We are committed to continual improvement and aspire to elevate our system to new heights in the future. For the development of our system, we opted for the utilization of an Artificial Neural Network (ANN), a sophisticated computational model inspired by the human brain. Boasting an impressive accuracy rate of 98.29%, our ANN excels in making highly precise predictions. This level of precision is paramount in ensuring the efficacy and reliability of our system, instilling confidence in its functionality and performance.

9. REFERENCES

9. REFERENCES

- https://www.kaggle.com/datasets/scodepy/customer-support-intent-dataset
- https://www.javatpoint.com/machine-learning-algorithms
- https://scikit-learn.org
- https://www.guru99.com/nltk-tutorial.html
- https://seaborn.pydata.org/tutorial.html
- https://www.datacamp.com/community/tutorials/wordcloud-python
- https://www.geeksforgeeks.org/python-stemming-words-with-nltk/
- https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/