**Student Appointment Chat bot**

**ABSTRACT:**

Chatbot technology is software that allows businesses to communicate with their consumers. This technology has grown at an unparalleled rate, sparking a slew of new ideas in the brains of innovators. The report explains the project's main theory, technique, and execution. The goal of this project is to figure out why UMKC decided to adopt a chatbot for college/university students. Colleges, institutions, and public venues have been closed as a result of large-scale lockdowns since the commencement of covid-19 in March 2020. Physical activity is becoming more difficult as a result of the newly developed strict social distancing standards. Distancing oneself from others is still necessary. To meet these objectives, the University must devise a system that ensures that the number of students at any one location does not surpass a certain threshold. Students can use the chatbot, which is an online media tool, to check the availability of slots and book them for leisure activities. The activities are divided into major groups. Instead of entering into the program and manually enrolling, the application will cater to UMKC students who wish to register slots based on the suggested leisure activities.

**INTRODUCTION:**

Chatbot is one of the quickest and most effective ways to communicate. The chatbot will communicate with customers and reply to any questions they may have. For example, a user can talk with the chatbot to check up available flights on a specific date and book them. One of the most beneficial features of the chatbot is that it helps website owners to save money by removing the need for a large number of employees and offering service at any time of day. Based on meaning analysis and Natural Language Processing, the chatbot will understand the user's texts and respond with the appropriate response (NLP).

We are constructing a Chatbot application for the University of Missouri Kansas City as part of this project. Booking an appointment with an academic supervisor and the availability of entertainment activity slots at the institution are also options. The team's initial focus was on creating the chatbot application to create a dataset and questions that students/faculty members would ask, as well as evaluating incoming texts and providing the desired response. The chatbot will greet the user and inquire about the service that can help them with. This software attempts to make it easier for students and professors to schedule their appointments.

**PROPOSED WORK:**

The chatbot's main function is to answer to student questions in the absence of a staff. The chatbot can be used in any web browser by students. The chatbot receives the student's query, analyses it, and then responds to the user with an answer. The questions are stated in the intent, which will be used to train the chatbot to recognize them.

**IMPLEMENTATION:**

Implementation part focuses on the software part of the chatbot, the intent, and the dataset developed to train the chatbot. The algorithm of the process is followed by the design of the system and the motive and goals addressed by the chatbot.

The data, which is appropriately processed to obtain a format understood by the machine, is required to design the chatbot. Because computers are unable to understand human-readable text, the text is converted into a machine-readable format. Word embedding techniques must be used to represent the words.

# Working Algorithm

Step 1) Select a data set, for which we need to develop a chatbot.

Step 2) Prepare the set of entities with the patterns and the responses.

Step 3) Install the required packages in python

Step 4) Train the chatbot on the dataset to understand the intent of the user

Step 5) Develop the GUI and integrating it with the bot Step 6) Execute the codes for the results

Step 7) Exit

# Pre-processing using NLP:

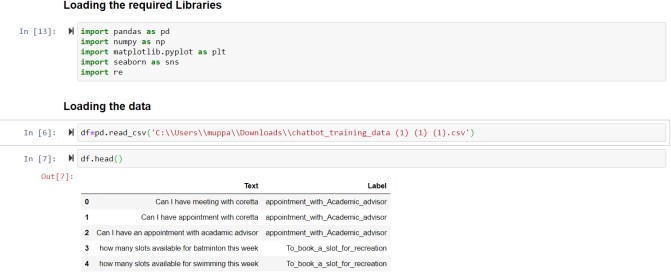
# The text is analyzed and pre-processed using NLP (Natural Language Processing) to clean it up and prepare it for further research by the model.

# NLP is a natural language processing toolkit that recognizes the components of text provided as input by the user at any given time. The NLP engine transforms the text language into a streamlined piece of data that the system can understand. The model's proposed chatbot is domain-specific and should be able to perform a variety of tasks.

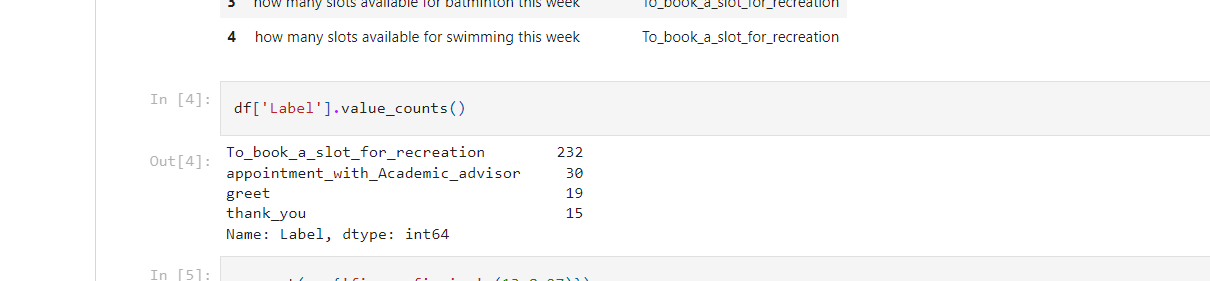
# Pre-processing of text using NLP model

# 

The coding for the backend of the chatbot application is performed in python. It includes many library functions like NLTK, TFIDF – count



vectorizer and Multiclassification model. The GUI is developed in python using Flask.



**label count**

# Splitting the data into train and test:

# To estimate the performance of the model created using machine learning algorithms, the train-test split evaluation is used. On test data, the algorithms are used to make predictions. The obtained results demonstrated the machine algorithm's performance in terms of prediction accuracy

The train dataset is used to fit the machine learning model. The test dataset is used to evaluate the performance of the machine learning algorithm

# 

**Splitting the train and test data**

# Extracting the features on the train and test data:

# Feature extraction is used to reduce the number of features in a dataset by creating new ones from the existing ones. The new set of features can be used to summarize the dataset's original features. The paper demonstrates the use of Count vectorizer for chatbot development.

# 

**Feature extraction**

In Python, the count vectorizer is a tool provided by the scikit-learns library. The frequency of the words in our data is used to transform the text into a vector, which is then used to train the classifier. The count vectorizer generates a matrix with columns for each unique word. The words aren't saved as strings; instead, they're assigned an index value.

# Selecting the algorithm:

# Context classification is a supervised learning algorithm used in machine learning. In supervised learning, the data fed to the network is labeled with the dataset's most important features. The model learns from the data by interpreting the labels to understand the most important aspects of the data, which are divided into distinct categories. Different classification algorithms, such as Decision trees, Nave Bayes, Gradient Boosting, and SVM, can be used to complete the task (support vector machine)

# Naive Bayes and Multinomial NB:

The chatbot's intent classification is based on a model created with Nave Bayes and Multinomial Nave Bayes classification.

Intent Classifier: This component takes user input, interprets it, and then connects it to the chatbot's supported intent.

Entity Extractor: This tool extracts critical information from a user's query.

# Fitting the train data set with model:

Each question and the response given to the user are labelled, and the dataset containing the user's intent and corresponding answers is created. The label connects the question and the response. The dataset contains multiple questions with the same response.

The fit() method is used to train the classifier by passing in the training vectors and labels. The user's input is collected using the chatbot on the backend, and the vectorizer is used to convert it into a vector. The model deduces the intent and uses the classifier to predict the label.

# 

**Model training**

**Chatbot Framework:  
UI creation for chat icon:**

For the UI, a theme CSS file is created, which defines the UI's overall look and feel. To render the bot icon in the web browser, the UI was created using html and JavaScript files.

The chatbot application framework is built by exposing this UI code as a flask API in Python that can be used to interact with the bot's backend, which includes the ML model. The ML model for the bot's operation is set up in the simple bot api UI project's config.js file.

The chatbot includes the ability to log the user's chat conversations as well as the code to push the necessary data into a SQL database. Deserializing the pickled ML model with the pickle library in Python is used to configure the ML model in the chatbot api project.

# Workflow of the Chatbot:

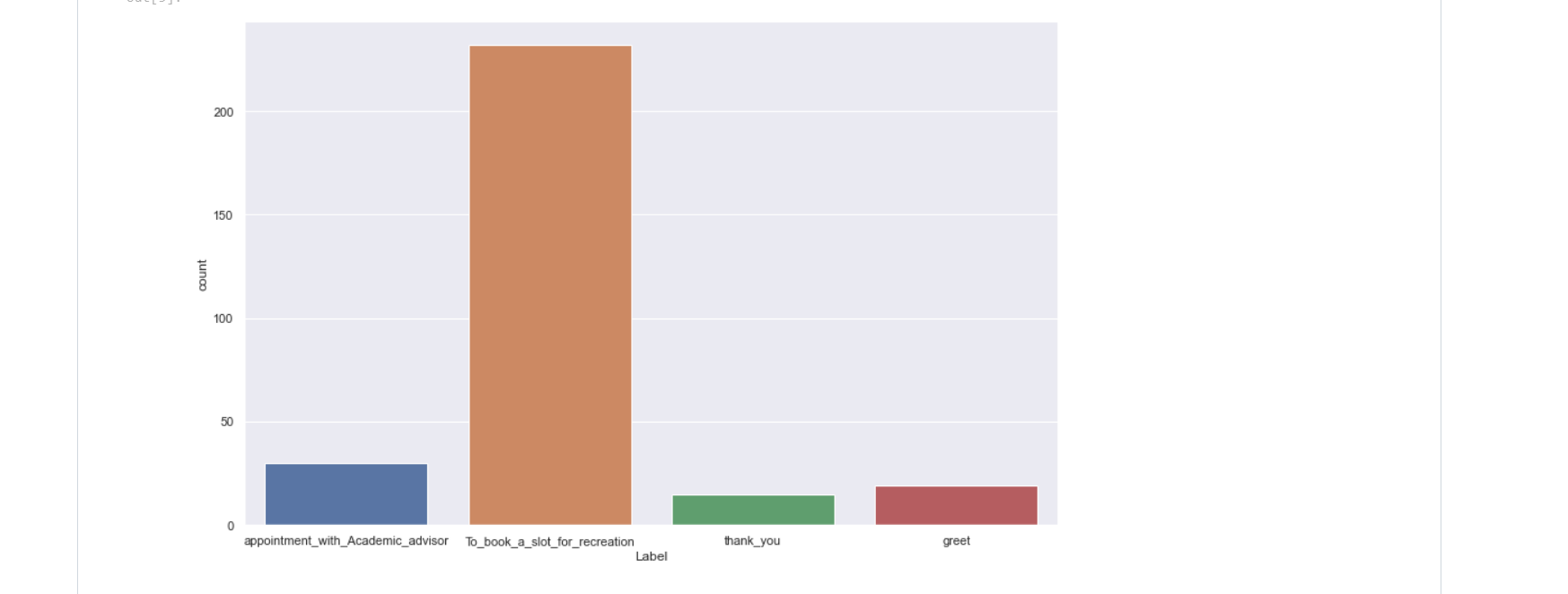
# 

**Workflow of the chatbot model**

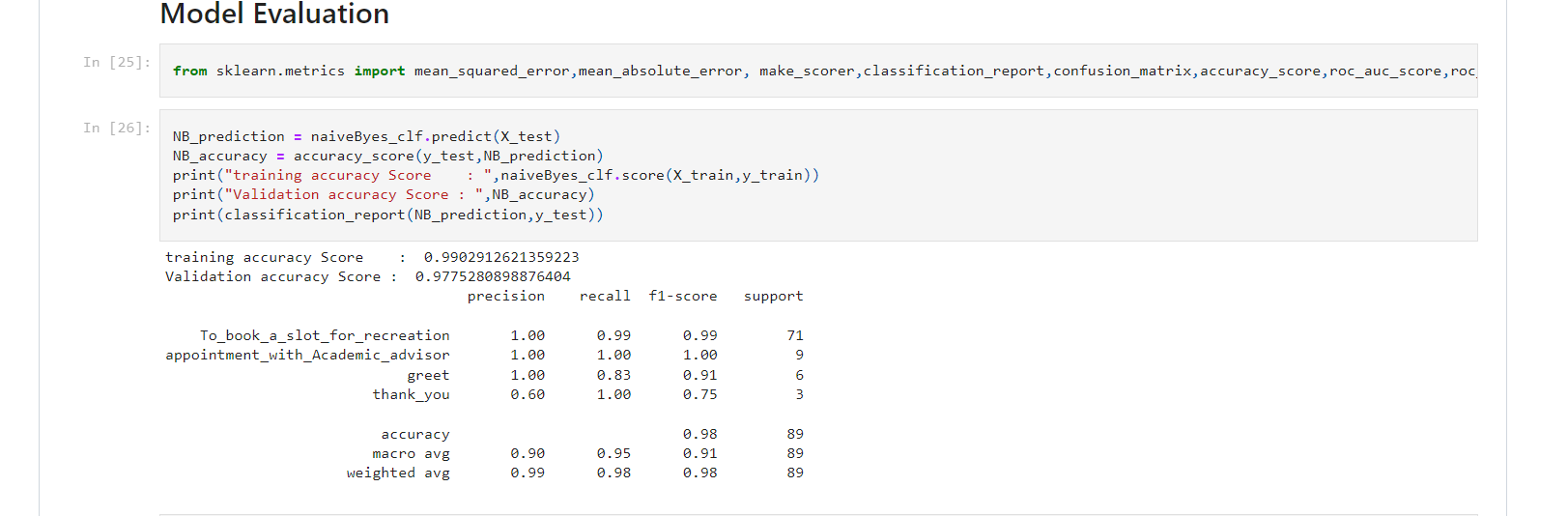
**EVALUATION AND RESULTS:**

**Dataset Statistics:**

The datasets and class distributions are analyzed using an exploratory data analysis approach. The class distribution in the dataset is summarized in the results analysis below.



# Evaluation matrix using accuracy metric and confusion matrix



**Confusion Matrix**

The model evaluation is used to check the trained model's accuracy on the test dataset. 99.03 percent accuracy was achieved. With 99 percent accuracy, the classifier can classify the user's intent.

# Deserialization of the built model:

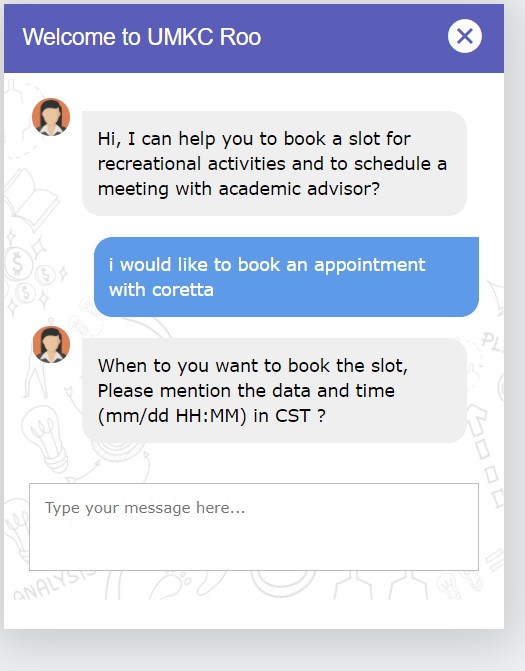
The pickle library from Python is used to serialize the ML model.



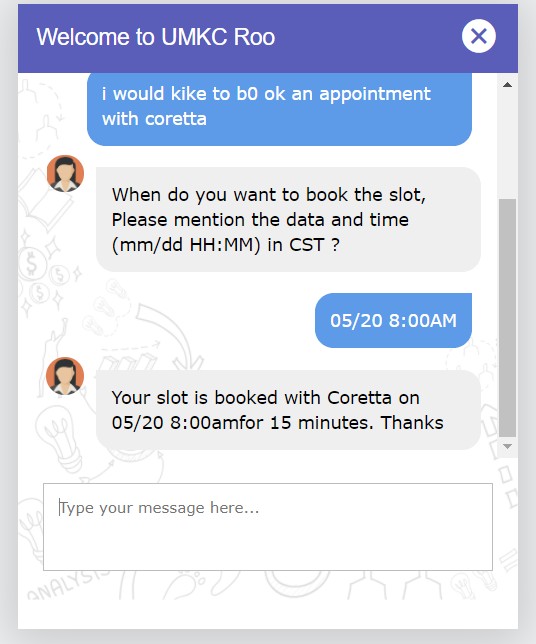
**Pickle library for serialization**

# Working Chatbot:

when the user logs into the web browser, the chatbot greets them with a greeting message. "Hello, I can assist you in scheduling a slot for recreational activities as well as a meeting with your academic advisor.

Depending on whether the user wants to book a slot for recreational activities or an appointment with an academic advisor, the chatbot can figure out what the user's intent is. The chatbot will respond to the user with a message asking for information about the time and date for booking the slot based on the user's intent.

**Chatbot responding to queries**



**CONCLUSION:**

It has a variety of applications, and we'll go over one of them below to show how useful the appointment booking chatbot can be for the university. The chatbot understands sentence structure using AI and natural language processing technology, which in this case is the questions the student will ask the chatbot. That information is processed by the chatbot, which improves its ability to answer questions over time. The data scientists' intent includes a variety of question combinations that the students will ask the chatbot. It could be making an appointment with an academic advisor or reserving slots for recreational activities, all of which are fed into the backend model. The chatbot deduces the user's intent and provides an answer based on the available data.

**REFERENCES:**

* R. Khan and A. Das, “Introduction to chatbots,” in Build Better Chatbots, Berkeley, CA: Apress, 2018, pp. 1–11.
* A. Følstad and P. B. Brandtzæg, “Chatbots and the new world of HCI,” Interactions. ACM.org, vol. 24, no. 4, pp. 38–42, Jun. 2017.
* A. Schlesinger, K. P. O’Hara, and A. S. Taylor, “Let’s talk about race: Identity, chatbots, and AI,” in Proc. the 2018 CHI Conference on Human Factors in Computing Systems (CHI ’18), 2018, pp. 1–14.
* A. Schlesinger, K. P. O’Hara, and A. S. Taylor, “Let’s talk about race: Identity, chatbots, and AI,” in Proc. the 2018 CHI Conference on Human Factors in Computing Systems (CHI ’18), 2018, pp. 1–14.
* C. J. N. J. S. S. Divya Madhu, "A novel approach for medical assistance using trained chatbot," in International
* Artificial Linguistic Internet Computer Entity. Retrieved

from [https://en.m.wikipedia.org/wiki/Artificial\_Linguistic\_Internet\_C](https://en.m.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity) [omputer\_Entity](https://en.m.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity)

* George Seif (2018, October 2). An easy introduction to Natural Language Processing..Retrieved

from <https://towarsdatascience.com/an-easy-introduction>-to-natural- language-processing--b1e280129c1

* Conference on Inventive Communication and Computational Technologies (ICICCT), 2017.
* S. A. N. H. Hameedullah Kazi, "Effect of Chatbot Systems on Student’s Learning Outcomes," SYLWAN, 2019.

**VIDEO LINK:**

<https://drive.google.com/file/d/1jsdVh85jyEyGpajK_FEX1pjdCPSVaiYj/view?usp=sharing>

**Team member names:**

Haritha Thumukuntla (htmrc@umsystem.edu)  
Sandeep reddy Chevula (scv7n@umkc.edu)  
Chenna Srinivas (sc7vc@umsystem.edu)  
Mamjala Manisha (mmg6m@umsystem.edu)