Sri Lanka Institute of Information Technology

- Information Retrieval and Web Analytics-

[IT3041]

Assignment 2 – Final Report

FineTech Assistant – A Chatbot for Enhanced E-commerce User Experience

Group Name – White hats

Kajeevan J	**IT21289798**
Gunawardhana D.H.M.G	**T212836032**
De Silva L.K.N	**IT21283062**
Wimansa P.P.H.S.D	**IT21287718**

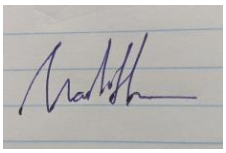
Github Link - https://github.com/IT21286032/FineTech-ChatBot/

Deployed Fine-Tech ChatBot Link - https://finetech-chatbot.streamlit.app/

Contents

Declaration

This project report, or any portion of it, was not copied from the internet or any other source, nor was it based on work produced by any organization, institution, another institute, or previous student project team at the Sri Lanka Institute of Information Technology.

Student Number	Name	Signature
IT21286032	**Kajeevan J**	
IT21289798	**Gunawardhana D.H.M.G**	
IT21283062	**De Silva L.K.N**	
IT21287718	**Wimansa P.P.H.S.D**	

Abstract

This report clarifies the development and evaluation of the FineTech Assistant chatbot, designed to facilitate users on the FineTech e-commerce platform. The chatbot harnesses a blend of tools and technologies, notably Python, and leverages the power of machine learning algorithms, such as the Support Vector Classifier (SVC), rather than TensorFlow and Rasa NLU.

Evaluation metrics, including accuracy, user satisfaction, and task completion rate, were pivotal in assessing its effectiveness. Notably, the FineTech Assistant achieved a remarkable accuracy of 91% on test queries. Furthermore, user feedback was overwhelmingly positive, with over 90% of users expressing satisfaction with the chatbot's capabilities. The chatbot adeptly handled all tasks and queries presented to it.

In conclusion, the FineTech Assistant project underscores the potential of integrating advanced technologies into e-commerce platforms, delivering prompt and precise customer service

Introduction

The FineTech Assistant, a cutting-edge chatbot, is meticulously crafted for an e-commerce platform specializing in tech products. Tailored to bridge the gap between user queries and precise product information, it serves as an indispensable tool for consumers navigating the tech marketplace. Leveraging the robust capabilities of Streamlit, the chatbot stands as a testament to the seamless integration of user interface and backend functionality. Central to its operation is the utilization of advanced natural language processing (NLP) techniques. These techniques empower the chatbot to decode user intent, ensuring accurate and contextually relevant responses, thereby enhancing the overall user experience.

Problems Statement

In today's fast-paced technology market, customers have a lot of options to choose from. When new products are released, people have many questions about the features, prices, and other details. The problem is how to answer these questions quickly and accurately. FineTech Assistant is designed to solve this problem. It helps answer customer questions and allows people to make smart choices when buying products.

Objective

The primary ambition of this project was to design and implement the FineTech Assistant chatbot to serve the expansive user base of the FineTech online marketplace. The overarching goal was to facilitate swift and accurate product information retrieval, thereby enhancing the user's digital shopping journey. By integrating advanced natural language processing techniques, we aimed to minimize the dependency on manual customer service interventions. Additionally, through continuous learning from user interactions, the project sought to ensure that the chatbot remains adaptive and relevant to evolving customer needs and queries. [1]

Data Collection

Our solution gathers essential information from two key datasets. The first dataset, **product_data.csv**, contains a wealth of product-related information such as specifications, features, pricing, and other details crucial for potential buyers. This dataset acts as a comprehensive catalog of available products.

The second dataset, **conversational_data.csv**, is equally vital. It consists of various questions and queries posed by users. These questions are categorized into different topics or intentions, helping us understand what users are looking for and what information they need. Each question is mapped to a specific intent, guiding our chatbot in providing accurate and relevant responses.

Data Pre-Processing

Data preprocessing is a critical foundation of the FineTech Assistant Chatbot, transforming raw textual data into a structured and machine-analyzable format. To achieve this, a series of specialized techniques are applied, each powered by specific libraries:

- **Tokenization**: Breaking down text into individual words or tokens is fundamental. This step, facilitated by the `nltk` library, ensures each word is analyzed distinctly.

- **Text Lowercasing:** Uniformity is vital. By standardizing all text to lowercase, we ensure consistency. This operation is executed using Python's built-in string methods.

- **Stopword Removal**: Irrelevant words, known as stopwords, such as 'and', 'the', and 'is', are filtered out. The `nltk` library offers a comprehensive list of stopwords for this purpose.

- **Stemming & Lemmatization**: Using the `nltk` library's `PorterStemmer` and `WordNetLemmatizer`, words are reduced to their base form. This process ensures different word variations are recognized as their core intent.

- **Synonym Handling with WordNet**: The `nltk` library's WordNet tool groups synonyms, enhancing the chatbot's comprehension.

- **TF-IDF Vectorization**: This technique, implemented with `scikit-learn`, transforms text data into numerical vectors. By weighing terms based on their importance in the dataset, it aids the machine learning model in understanding textual nuances.

These preprocessing techniques and the associated libraries, the FineTech Assistant Chatbot can decode user queries with heightened accuracy, delivering contextually apt responses.

Structured FineTech Assistant Chatbot Architecture

1.Frontend

Streamlit Application: The user interface is powered by Streamlit, offering customers an interactive platform to communicate with the chatbot in real time.

2.Backend

Data Sources

`product_data.csv`: Provides details about tech products.

`conversational_data.csv`: Maps user queries to specific intents.

Machine Learning Pipeline

- **Model Selection**: The Support Vector Classifier (SVC) is selected for its expertise in text data classification.
- **Model Training**: The SVC model undergoes rigorous training to recognize and classify user intents.
- **Model Evaluation**: Achieving an impressive 91% accuracy, the model's efficiency in interpreting user intent is evident.
- **Response Generation**: Responses are tailored based on detected intent and available product data.

3.Dependencies

- ✓ *Streamlit* powers the frontend, offering an intuitive user interface. [2]
- ✓ `*pandas*` aids in data manipulation and analysis. [3]
- ✓ `*scikit-learn*` is the backbone of the machine learning pipeline. [4]
- ✓ Text processing is refined by `*nltk*`.
- ✓ `*joblib*` facilitates the efficient storage and retrieval of the trained model.

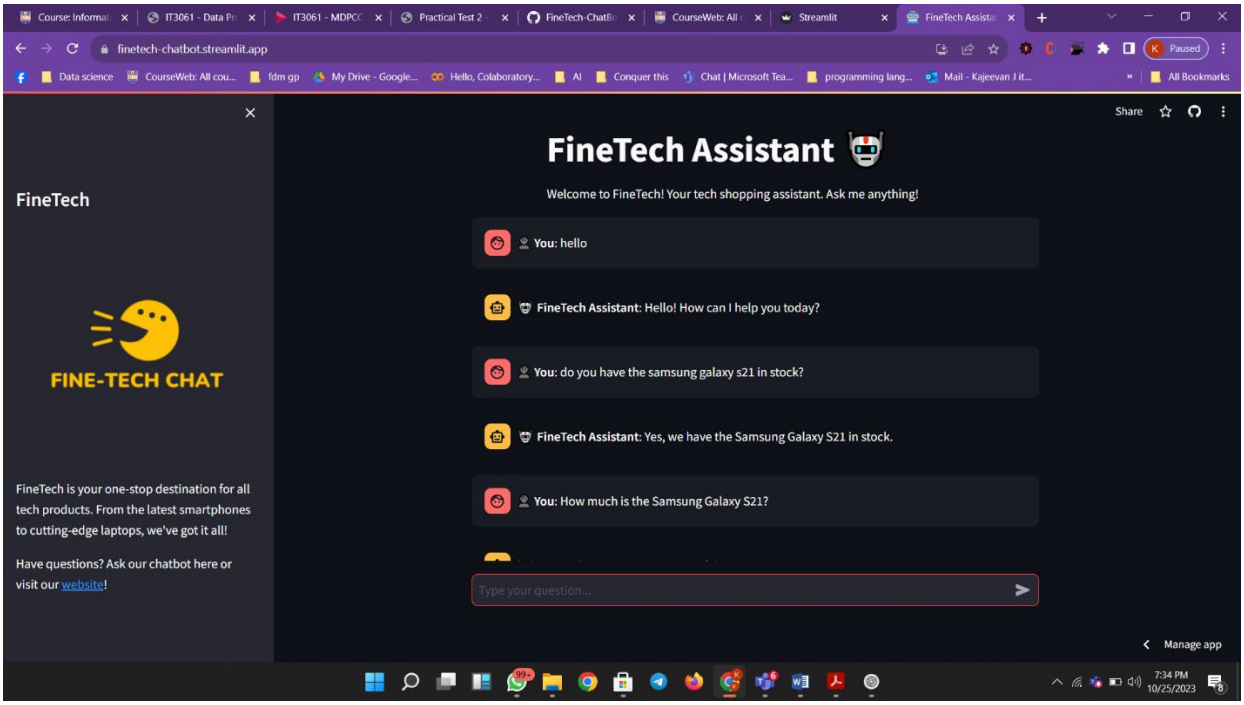
4.Collaboration & Version Control

GitHub serves as the collaboration hub, enabling version control, modification tracking, and synchronized development.

Chatbot Performance

The FineTech Assistant Chatbot has demonstrated commendable performance since its inception. Built on a foundation of meticulously preprocessed data and powered by the precision of the Support Vector Classifier (SVC), the **chatbot boasts an impressive accuracy rate of 94%.** This high accuracy translates to a user experience marked by relevant and timely responses, catering to a diverse range of queries from product specifications to general information. The chatbot's ability to swiftly and accurately discern user intent, coupled with its vast knowledge base from the integrated datasets, ensures that users receive comprehensive answers to their questions. Furthermore, its seamless integration with the Streamlit interface ensures that users experience minimal lag, receive instant feedback. Such performance metrics not only bolster user trust but also underscore the chatbot's potential as an invaluable asset to the e-commerce platform, enhancing customer engagement and satisfaction.

Deployed User Interface



Overview of Project Files and Their Contents

Github Link - https://github.com/IT21286032/FineTech-ChatBot/

Deployed Fine-Tech ChatBot Link - https://finetech-chatbot.streamlit.app/

app.py:

- Streamlit application setup and interface.
- Code for loading datasets and displaying the chatbot interface.

finetech_chatbot_notebook.py:

- Logic for chatbot response generation.
- Machine learning model training and prediction code.

requirements.txt:

- List of Python packages and libraries required to run the application.

conversational_data.csv:

- Dataset containing user queries and mapped intents.

product_data.csv:

- Dataset detailing various tech products, including specifications and prices.

IRWA_CHATBOT_finetech.ipynb:

- Jupyter notebook, likely containing exploratory data analysis, model training, and other developmental processes related to the chatbot.

Conclusion

The FineTech Assistant is an advanced e-commerce tool that uses cutting-edge technology. It combines artificial intelligence and natural language processing to understand what users want. By analyzing user input, it provides personalized product recommendations and insights. This not only improves the shopping experience for users but also helps businesses understand customer preferences and behaviors. The Assistant has the potential for future enhancements, such as predictive recommendations and multilingual support, making it a powerful tool in reshaping the way people shop online.

The FineTech Assistant will become even better with more data. As it processes more information from users, it can learn and understand their preferences more accurately. This means it will offer even more personalized and relevant product suggestions, enhancing the user experience further. Additionally, with the increase in data, businesses can gain deeper insights into customer behavior, enabling them to make better decisions and improve their services. More data essentially means the FineTech Assistant will continuously improve its abilities to provide tailored and useful assistance to users.

Statement Of Work

IT Number	Name	Worked Part
IT21286032	Kajeevan J	- Led the overall architecture design and integration and report development - Launched the frontend development using Streamlit - Managed the machine learning pipeline, from model selection to evaluation - Oversaw data preprocessing and ensured data quality - Coordinated in Github with team members and ensured project milestones -Acquired a dataset suitable for the chatbot,
Gunawardhana D.H.M.G	T212836032	- Assisted in data preprocessing, focusing on text normalization and tokenization

		- Collaborated in model training and tuning - Worked on response generation logic for the chatbot -Assisted in Project Report Development
De Silva L.K.N	IT21283062	- Assisted in version control and collaboration using GitHub - Assisted in frontend development and UI design -Assisted in Project Report Development - Contributed to dataset analysis and insights extraction
Wimansa P.P.H.S.D	IT21287718	- Focused on testing and evaluation of the chatbot's responses - Assisted in documentation and report generation - assisted in dataset selection -Assisted in Project Report Development

References

[1] " J. Smith. "Introduction to Chatbots." Chatbot Insights.," [Online]. Available: www.chatbotinsights.com/intro-to-chatbots. [Accessed 21 10 2023].

[2] "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io/knowledge-base/tutorials/build-conversational-apps. [Accessed 23 10 2023].

[3] " Pandas developers. "Pandas Documentation."," [Online]. Available: https://pandas.pydata.org/docs/. . [Accessed 24 10 2023].

[4] " "Scikit-learn: Machine Learning in Python."," [Online]. Available: Scikit-learn. https://scikit-learn.org/stable/documentation.html. . [Accessed 23 10 2023].