Innovation Showcase Report: Integrating Data Science Models Into Applications

GROUP - AMISHA MEKA 001362259
LAXMI PRANAVI MALLEKEDI 001356134
SRI LAHARI KATLA 001361016
VAISHNAVI TAPETLA 001327699

Table of Contents

Introduction

The Arch Data Network Innovation Showcase presents an extraordinary challenge to engineering students creating a novel data science model-based application for students this is an immense opportunity to participate in data driven innovation. This report explains the design and development of a Recipe Suggestion Chatbot that employs Natural Language Processing along with Classifcation models. Its primary objective is to recommend recipes to users in relation to the components they possess, adding a level of personalization, interactivity, and user participation by attractive the front-end interface. It is a requirement for the Innovation Showcase that the application incorporate a data science model which processes the user input and generates meaningful outcomes, hence the project was undertaken.

Project Objective

The overarching goal for this project is to develop an application that implements a data science model, with a special importance on Natural Language Processing (NLP) for word comprehension and command execution and classification models that arrange recipes into classes according to the level of difficulty or preparation time (Rajwal et al., 2025). This project make parallel with the Innovation Showcase supplies, as it includes:

1. A user-friendly interface
2. A front-end system of a data science model that offers output recipe suggestions based on user input.
3. A backend API integrating the data science model and the frontend.

Application concept

The chatbot will be a Recipe Suggestion Chatbot which asks users for elements they have on hand and suggests relevant recipes (Hamideh Kerdar et al., 2025). It will incorporate the following core features:

1. The Chatbot Requests that Users Input Ingredients: The Chatbot asks users to type in the ingredient list of what they have.

2. Recipe Suggestion: Using natural language processing (NLP) and ingredient recognition, the chatbot fetches the relevant recipes from the database.

3. Recipe Classification: Each recipe is classified based on preparation time and difficulty into easy, medium, or hard.

4. User Interaction: Users can narrow their search based on dietary restrictions (vegetarian, gluten-free, etc.) if they so wish.

This application provides an interactive front-end and applies a data science model to security a customized user experience.

Data Science Model Selection

Natural Language Processing Model

The selected data science model for the application is Natural Language Processing(NLP) model which will enable the chatbot to know and process the ingredients provided by the user. The application will utilize NLP for recognizing ingredients from the user provided text input, enabling matching of ingredients with the recipes stored in the database (Mzwri and Turcsányi-Szabo 2025).

Classification Model

Besides that, a classification model will sort the recipes according to their difficulty level (easy, medium, hard) or preparation time. This model will be trained on predefined data to classify

certain recipe difficulty patterns so that it can classified them correctly.

Model Architecture

1. Preprocessing: The input data will be subject to preprocessing. Tokenization would be performed as well as stop-word elimination and lemmatization is done.

2. Ingredient Recognition: The chatbot will be able to interact the user using the NER methods which allows a person to save a particular use to abstract words such as ”chicken”, ”tomatoes”, and ”spinach”.

3. Recipe Retrieval: After determining the ingredients, the application will try to match the elements with recipes in the database using techniques such as TF-IDF or Word2Vec. The best matches will be picked based on semantic similarity.

4. Recipe Classification: Once the recipes have been retrieved, a classification model will categorize them based on particular attributes, such as the level of difficulty and preparation time. A decision tree or another classification algorithm will be used to assess the recipe difficulty.

Training the Model

The model will be trained using a dataset that comprises recipes with associated elements, step-by-step instructions, and designated levels of difficulty. This data will be collected from public recipe APIs or websites. The dataset will be cleaned and preprocessed before training, focusing on identifying ingredients, categorizing recipes, and creating suggestions tailored to user profiles.

Model Evaluation

The model will be assessed using accuracy methods, including precision, recall, and F1-score. Cross-validation will be implemented to test the model reliability and capability to handle before unencountered data. A set of data separate from the training dataset will be used to assess the model performance in practical situations.

Application Front-End Design

User Interface (UI)

The application front-end will be developed to ensure optimal user interaction throughout the application. The user will be able to engage with the chatbot using a classy interface with which they can submit their ingredients (Shethiya, 2025). The design will combine the following components:

1. Input Field: Users will enter their ingredients using a text box.

2. Recipe Suggestions: The chatbot will provide suggestions for recipes that include the name of the recipe, the ingredients, the orders to be followed, and the estimated difficulty level.

3. Refinement Options: Users will be able to filter the suggestions based on restrictions, such as vegetarian or gluten-free.

4. Interactive Buttons: There will be buttons for "Get New Suggestions" or "Refine Search" to enhance the user interaction further.

The front-end will be written in React since it allows easy updates to user interfaces. The application interface will be done with Tailwind CSS which will enable style customization for responsiveness and flexibility.

Back-End Design

Flask, a Python based web framework is chosen for development of the back-end since it is lightweight and easy to use for building APIs. The Flask application will manage processing of user inputs, interaction with the NLP and classification models, and the communication with front-end of the application (Shen, 2025).

1. API Development: The backend will provide an API that the frontend will utilize to interact with system functions like processing user input and retrieving information from the recipe database.

2. Model Deployment: The server will host the trained NLP and classification models and process input text to suggest appropriate recipes.

3. Database: A recipe database will maintain records of selected recipes containing their ingredients, steps, and level of difficulty. The backend will query this database to provide relevant recipes based on ingredients stored by users.

Challenges and Solutions

Challenge 1: Input Ingredients Ambiguity

Users can type ingredients in different forms which leads to semiformal doubt. One user could provide "tomato" while a different user provides "cherry tomatoes". This issue needs to be solved to ensure the smooth functioning of a chatbot (Humm et al., 2025).

Solution:

- **Ingredient Normalization:** The team will carry out an constituent normalization procedure that combines different phrases and differences of words.
- **Spell Correction:** An algorithm will be implemented to improve the spelling accuracy of listed ingredients, particularly focusing on potential errors or misspelled terms.
- **Synonym Mapping:** The group will develop a system which performs synonym mappings to identify multiword names and various terms for fixings. This allows for understanding of local differences in naming components which lowers doubt even more.

Challenge 2: Merging Ingredients

The main problem lies in the merging of elements. This attitudes a challenge by trying to fulfill the user description check as some ingredients may not perfectly align with those in the recipe database (Dakhia et al., 2025).

Solution:

- **Semantic Matching:** To pair ingredients with recipes, the team will use semantic comparison techniques such as TF-IDF or Word2Vec.

- **Fallback Mechanism:** A fallback mechanism will suggest formulas using partial matches of the ingredients if no particular matches are available.
- **Contextual Ingredient Grouping**: The team will also use contextual ingredient grouping to merge specific items, for instance, potatoes and sweet potatoes. This approach will help in accepting changes in ingredient forms while still retrieving relevant recipes. Moreover, it will allow the chatbot to suggest more instructions even when exact components are unavailable. This will provide users with more diverse suggestions irrespective of the inputs provided.

Expected Outcomes

The project outcome should produce a complete working Recipe Suggestion Chatbot that is capable of:

1. User input regarding components will be processed and identified through NLP.
2. Recipes will be classified in terms of difficulty or preparation time.
3. The chatbot will use recipe fixings provided by the user to suggest relevant recipes.
4. The system will have an communicating front-end interface which will improve the user experience.

The application will help as a proof of concept on the use of NLP and classification models in real life, which will be critically beneficial for the team learning. Furthermore, the project will serve as an example of how a data science model can be developed and subsequently integrated into an end-to-end application (Hamideh Kerdar et al., 2025).

Conclusion

This document describes the construction of a Recipe Suggestion Chatbot powered by a data science model that processes user queries and generates recipe suggestions based on the input. The project is in complete compliance with the goals of the Innovation Showcase as it integrates concepts of data science and user interaction to develop an operational application. The application, built with Natural Language Processing (NLP) and classification models, aims to engage the user in an interactive experience, meeting the user expectations and the requirements

of the assignment for additional credit. This solution not only creates the practical use of data science in real-world applications but also improves the user knowledge by offering personalized suggestions.

References

Dakhia, Z., Russo, M. and Merenda, M., 2025. AI-Enabled IoT for Food Computing: Challenges, Opportunities, and Future Directions. *Sensors*, *25*(7), p.2147.

Hamideh Kerdar, S., Kirchhoff, B.M., Adolph, L. and Bächler, L., 2025. A Study on Chatbot Development Using No-Code Platforms by People with Disabilities for Their Peers at a Sheltered Workshop. *Technologies*, *13*(4), p.146.

Hamideh Kerdar, S., Kirchhoff, B.M., Adolph, L. and Bächler, L., 2025. A Study on Chatbot Development Using No-Code Platforms by People with Disabilities for Their Peers at a Sheltered Workshop. *Technologies*, *13*(4), p.146.

Humm, B.G., Blum, L.B., Busse, J., Decher, S., Dembach, M., Goetz, C., Hoppe, T., Kaupp, L., Schwerk, A., Seidlmayer, E. and Siegel, M., 2025. New trends in applied machine intelligence: Selected results of the Dagstuhl Workshop AMI'2024. *Informatik Spektrum*, pp.1-17.

Mzwri, K. and Turcsányi-Szabo, M., 2025. Internet Wizard for Enhancing Open Domain Question Answering Chatbot Knowledge-Base in Information Seeking. In *Chatbots and Mental Healthcare in Psychology and Psychiatry* (pp. 147-180). IGI Global Scientific Publishing.

Rajwal, S., Zhang, Z., Chen, Y., Rogers, H., Sarker, A. and Xiao, Y., 2025. Applications of Natural Language Processing and Large Language Models for Social Determinants of Health: Protocol for a Systematic Review. *JMIR Research Protocols*, *14*(1), p.e66094.

Shen, T., 2025. Research on the Design of Data Asset Valuation Platform Based on WEB. *Open Access Library Journal*, *12*(4), pp.1-12.

Shethiya, A.S., 2025. Scalability and Performance Optimization in Web Application Development. *Integrated Journal of Science and Technology*, *2*(1).