|  |  |
| --- | --- |
| Student no. | Initials and Surname |
| 218069413 | C Molelekoa |
| 222312971 | A Ratsoana |
| 218552785 | P Nyembe |
| 222329025 | SM Khanya |
| 222355638 | NS Mnisi |
| 221278214 | B Gabiya |
| 216038154 | L Hlongwane |

SCHOOL AI RECOMMENDATION SYSTEM (EDU\_PORTAL)

https://github.com/ChabeliM-web/The-Big-8.git

AI Solution - DOCUMENTATION ASPECT

AI Solution

This AI Solution project is designed to help people find the best-fit schools in Gauteng if they’re looking to enroll. Using artificial intelligence, it carefully examines various details about each school—such as academic performance, location, grade level, and feedback from others—to provide personalized recommendations. By applying techniques like natural language processing (NLP) to interpret what users are looking for, and using graph neural networks (GNNs) to understand connections between schools, districts, and users, the system offers recommendations that are relevant and insightful. The goal of this project is to simplify the school selection process, making it easier and faster for users to find schools that meet their needs.

Problem Definition

In Gauteng, individuals seeking information on schools face a prolonged and often challenging process due to the large number of options available. Prospective students and parents must sift through extensive data to find schools that align with their educational needs, preferences, and specific requirements. This makes the enrollment process time-consuming and can lead to information overload, often hindering informed decision-making. Given the vast selection of schools across various districts, users require an efficient, accurate, and user-friendly system to streamline their search and ensure they access relevant and personalized information.

This problem is highly relevant to the theme, as it highlights the need for improved access to school information and simplified decision-making for educational enrollment. By addressing this, an AI-powered recommendation system can significantly enhance the user experience, providing fast, reliable, and tailored school suggestions. This solution not only aligns with the theme of making information more accessible and actionable but also benefits users by reducing the time and effort required to identify suitable schools. It will simplify the enrollment process, allowing users to focus on key factors that matter most to them and ensuring a more straightforward path to choosing the right school.

Business Objectives

**Business Objectives for EduPortal**

EduPortal aims to revolutionize school selection and enrollment for South African families by leveraging AI technology to provide a seamless, personalized experience for users looking to find the best-fit schools in Gauteng. The following business objectives define our goals for EduPortal’s success, grounded in specific, measurable outcomes that align with South African educational needs:

1. **User Acquisition and Platform Adoption**
   * *Objective*: Reach at least 100,000 registered users within the first year of EduPortal’s launch, with a focus on families across Gauteng looking to streamline school selection.
   * *Success Criteria*: Achieving at least 10,000 active monthly users and receiving a user satisfaction rating of 4.5/5 through feedback surveys.
   * *Relevance*: This objective ensures EduPortal is widely adopted and positively received, proving its value and accessibility to South African families.
2. **Revenue Generation through Subscription and Partnership Models**
   * *Objective*: Generate R2 million in revenue in the first year by offering premium features for a subscription fee, including advanced recommendations, priority support, and exclusive resources.
   * *Success Criteria*: Secure partnerships with at least 50 educational organizations, service providers, and government bodies that contribute to EduPortal’s revenue and influence in the education sector.
   * *Relevance*: Creating a sustainable revenue model supports EduPortal’s long-term growth and enables continuous improvements to the platform, benefiting users while driving profit.
3. **Educational Impact and User Accessibility**
   * *Objective*: Improve user access to critical school information by reducing the average time needed to find relevant school options by 70% compared to traditional methods.
   * *Success Criteria*: Enable access to detailed information for over 90% of Gauteng’s schools within the platform, with data accuracy verified quarterly.
   * *Relevance*: Ensuring broad access to accurate school data within Gauteng makes EduPortal a trusted resource for families, allowing them to make informed decisions based on up-to-date and comprehensive information.
4. **Continuous Improvement and AI Model Enhancement**
   * *Objective*: Improve the AI model’s recommendation accuracy to achieve at least 85% satisfaction from users regarding school suggestions and relevance.
   * *Success Criteria*: Implement quarterly AI model updates based on user feedback and data analysis to refine the accuracy and relevancy of recommendations.
   * *Relevance*: Regularly improving EduPortal’s AI model ensures that it stays aligned with evolving user needs and maintains a high standard of service, which is crucial for long-term success.

**Requirements, Constraints, and Risks**

1. **Requirements**
   * Develop a robust, AI-powered recommendation system that integrates natural language processing (NLP) and graph neural networks (GNNs) to understand and match user preferences with suitable schools.
   * Ensure secure handling of user data in compliance with South African data protection regulations (e.g., POPIA).
   * Partner with local educational institutions and government bodies to maintain data accuracy and expand access to educational resources.
2. **Constraints**
   * The platform must remain accessible on low-bandwidth internet to reach all areas in Gauteng, especially in underserved communities.
   * Operate within a limited initial budget (R5 million) allocated for platform development, marketing, and user acquisition in the first two years.
3. **Risks**
   * **Data Accuracy Risks**: Inaccurate or outdated information about schools could mislead users and affect their experience.
   * **Privacy and Security Risks**: Handling sensitive user information requires rigorous data protection measures to prevent breaches and ensure compliance.
   * **Adoption Risks**: Lack of adoption in target user groups could undermine EduPortal’s objectives, making it essential to invest in marketing and user engagement strategies tailored to South Africa.

**Tools and Techniques**

1. **Natural Language Processing (NLP)**: Used to analyze user queries and understand preferences, enabling the recommendation system to offer more personalized results.
2. **Graph Neural Networks (GNNs)**: Applied to model complex relationships between schools, districts, and user demographics, enhancing recommendation accuracy.
3. **Cloud Infrastructure**: A scalable cloud platform (e.g., AWS or Microsoft Azure South Africa) to support EduPortal’s backend, ensuring reliability and efficient performance.
4. **Data Security Tools**: Advanced encryption and user data protection measures to comply with POPIA and build trust with users.

By focusing on these objectives and leveraging AI-driven tools, EduPortal aims to transform the school enrollment process in Gauteng, providing measurable value to users and aligning with the long-term educational goals of South Africa.

AI Solution - THEORETICAL ASPECT

Machine Learning Approach

To build an intelligent and responsive recommendation system for EduPortal, we are utilizing a robust Machine Learning (ML) approach that combines multiple advanced techniques to deliver precise and personalized school recommendations. The approach is strategically designed to meet the platform’s requirements of accuracy, scalability, and relevance to user needs.

**1. Natural Language Processing (NLP) for Query Understanding**

* *Objective*: Process and interpret user queries effectively to understand preferences and contextual needs, such as specific school attributes, location, grade levels, or performance metrics.
* *Approach*: We employ NLP techniques to analyze both the structured and unstructured data contained in user queries and reviews. For example, BERT (Bidirectional Encoder Representations from Transformers) is used as the underlying NLP model, enabling EduPortal to capture the nuances of user language. By analyzing user preferences in real-time, the model can distinguish between different query types (e.g., "high-performing primary schools in Johannesburg" vs. "affordable high schools near Pretoria").
* *Implementation*: After tokenizing and embedding user input, EduPortal’s NLP pipeline identifies key entities (e.g., school type, location) and sentiment in user reviews to provide meaningful recommendations. This ensures that the system accurately interprets user intent, laying the foundation for a personalized recommendation process.

**2. Graph Neural Networks (GNNs) for Relationship Modeling**

* *Objective*: Understand and model complex relationships between schools, districts, users, and other features to enhance recommendation relevance and accuracy.
* *Approach*: EduPortal’s dataset comprises structured data points, including school information, geographic locations, and user profiles. GNNs are chosen as the primary algorithm to represent these relationships due to their capability to capture dependencies within graph-structured data. By representing schools, districts, and user interests as nodes, and their interactions as edges, GNNs allow us to understand how various features interconnect, thus enhancing the recommendation engine’s accuracy.
* *Implementation*: We build a multi-layer GNN model that processes this network data, iteratively aggregating information from neighboring nodes (e.g., schools in the same district or with similar performance metrics) to make recommendations that are both contextually and geographically relevant. The model is trained to predict schools that would be of interest to a particular user, considering similarities in previous user profiles and historical enrollment patterns.

Data

To ensure EduPortal’s recommendations are accurate and contextually relevant, we used a dataset sourced from the Gauteng Department of Education. This dataset includes essential information about schools across Gauteng, which allows our AI models to deliver personalized recommendations effectively. The dataset is structured to contain multiple data points, each serving a specific purpose in the recommendation process:

1. **National EMIS Number**
   * A unique identifier assigned to each school, used to accurately track schools and avoid duplication in the system.
2. **Name of School**
   * The official name of each school, providing users with clarity and recognition when viewing recommendations.
3. **School Phase**
   * Specifies the educational phase (e.g., primary, secondary), helping the recommendation engine match user needs with the appropriate school level.
4. **Address of School**
   * Location data allows the platform to offer geographically relevant recommendations based on user proximity preferences.
5. **Education District**
   * Indicates the district in which each school is situated, enabling the platform to consider local educational policies, resources, and district-specific attributes in recommendations.
6. **Performance Percentage**

* Indicates the district in which each school is situated, enabling the platform to consider local educational policies, resources, and district-specific attributes in recommendations Provides insight into the school’s academic performance, aiding users in selecting schools based on quality indicators.

1. **Contact Details**

* Includes phone numbers or emails, allowing users to directly reach out to schools for additional information or inquiries.

Each of these data points plays a critical role in ensuring that EduPortal’s recommendations are precise and cater to user-specific needs, whether they are seeking proximity, specific phases, or other school attributes. This data forms the backbone of our AI-powered recommendation system, allowing it to deliver personalized, insightful suggestions that meet the needs of families throughout Gauteng.

Model

EduPortal utilizes a **Graph Neural Network (GNN)** model, supported by **Natural Language Processing (NLP)** for user query analysis. This combination enables the platform to provide highly relevant school recommendations based on user preferences, school data, and relational information within the education ecosystem in Gauteng.

**Model Evaluation for Accuracy**

To ensure the GNN model's recommendations are accurate, EduPortal uses the following evaluation methods:

1. **Precision and Recall**
   * *Purpose*: Precision measures the relevance of recommended schools by evaluating the percentage of suggested schools that users find useful, while recall ensures that the model doesn’t miss relevant schools in its suggestions.
   * *Method*: We calculate Precision@k and Recall@k (where "k" is the number of top recommendations presented) to understand how well the model identifies and ranks relevant schools for users.
2. **User Satisfaction Score**
   * *Purpose*: Direct user feedback is critical to evaluating overall model performance and recommendation relevance.
   * *Method*: EduPortal collects user satisfaction scores (on a 1-5 scale) for each recommended school. An average score of 4.0 or above indicates that the recommendations are meeting user expectations.
3. **Retraining and Continuous Improvement**
   * *Purpose*: To keep the model aligned with user needs and update it for new schools or user trends.
   * *Method*: Based on feedback and evaluation results, the model is retrained periodically to improve accuracy, incorporating new user interactions and updated school data.

Solution Techniques

**1. Graph Neural Networks (GNNs) for Relationship Mapping**

GNNs model relationships between schools, districts, and users, capturing complex connections to make recommendations contextually relevant.

**2. Natural Language Processing (NLP) for Query Interpretation**

NLP interprets user preferences in queries, ensuring the platform accurately understands and responds to specific needs.

**3. Collaborative Filtering for Personalization**

Collaborative filtering tailors recommendations by analyzing user behavior, making suggestions more aligned with individual preferences.

**4. Hybrid Recommendation System for Cold Start Problem**

A hybrid approach combining content-based and collaborative filtering addresses limited data situations, ensuring accurate recommendations for new users or schools.

**5. Continuous Monitoring and Feedback Integration**

User feedback and model performance metrics guide regular updates, keeping the system accurate and responsive to evolving user needs.

Natural Language Processing and Speech Synthesis

1. **Natural Language Processing (NLP)**
   * NLP enables EduPortal’s chatbot to understand user queries, allowing for personalized school recommendations. It processes a wide range of natural language inputs, ensuring the chatbot can respond accurately to diverse user needs.
2. **Speech Synthesis**
   * Speech synthesis provides an auditory response, making EduPortal more accessible to users who prefer voice interaction or have limited literacy. Text-to-speech (TTS) technology ensures clear and engaging spoken responses, enhancing user experience.

Deep Learning

**Graph Neural Network (GNN) Learning**

* **How it Learns**:  
  A GNN processes graph-structured data, where nodes represent entities such as schools, districts, and users, while edges represent relationships or interactions. The model learns by aggregating information from neighboring nodes in the graph to understand the features of each node. This allows the GNN to identify relationships and patterns relevant to school recommendations.
* **Training Process**:  
  During training, the GNN receives data such as school performance metrics, user preferences, and district characteristics. It uses this data to predict relevant schools for users. The model improves its predictions by adjusting its parameters through backpropagation, minimizing errors in its predictions.

**2. Natural Language Processing (NLP) Learning**

* **How it Learns**:  
  NLP processes textual data, such as user queries, reviews, and school descriptions. Using models like **BERT** (Bidirectional Encoder Representations from Transformers), the system learns to interpret context, intent, and specific details in text, such as identifying school-related keywords and sentiments.
* **Training Process**:  
  NLP models are trained using large, labeled datasets, where the model learns to predict the meaning of unseen text inputs based on the patterns it has learned. The system refines its ability to handle complex queries by adjusting its parameters using supervised learning methods, allowing for improved understanding and response generation.

**3. Learning from Datasets**

* **How it Learns**:  
  The AI learns from both structured data (e.g., school performance statistics, location) and unstructured data (e.g., user reviews and queries). The datasets provide key insights into school characteristics and user preferences, helping the system identify correlations and patterns relevant to recommendations.
* **Training Process**:  
  The system trains using labeled data, such as user ratings or school features, applying **supervised learning** techniques to predict the suitability of a school for a particular user. It may also use **unsupervised learning** to discover hidden patterns, such as clustering similar schools or predicting user preferences based on historical interactions.

**4. Iterative Feedback Loop**

* **How it Learns**:  
  As the AI interacts with users, it collects feedback, such as ratings, clicks, or satisfaction scores. This feedback is used to refine the recommendations and improve the overall accuracy of the system.
* **Training Process**:  
  The system incorporates this user feedback into its model, retraining periodically to improve its understanding of user needs and preferences. This ongoing feedback loop allows the AI to adapt to changing user behavior, making the recommendations more accurate and personalized over time.

Bottom of Form

Chatbot/Softbot

The **Chatbot** is a core feature of EduPortal, designed to enhance user interaction by providing immediate, personalized responses to user queries regarding school recommendations, details, and other educational services. This feature is both relevant and essential to the solution, as it offers a convenient, user-friendly interface that aligns with EduPortal’s goal of simplifying the school recommendation process. Below is a detailed explanation of how the chatbot is well-planned and appropriately set up for EduPortal:

**1. Relevance to the Solution**

The chatbot is central to providing a seamless and interactive experience for users seeking school recommendations or information. It helps bridge the gap between users and the vast amount of data contained in the EduPortal system. Users can interact with the chatbot to ask questions about schools, inquire about admission processes, learn about school performance, or even receive personalized school recommendations based on their preferences.

The chatbot uses Natural Language Processing (NLP) to understand and respond to user queries. This makes it highly relevant to EduPortal's goal of offering a simple, efficient way for users to access school-related information, without needing to manually sift through large amounts of data.

**2. Well-Planned and Appropriately Set Up**

The chatbot is designed with the following key features that ensure it is well-planned and effective:

* **User-Centric Design**:  
  The chatbot is designed to cater to the user’s needs by providing direct, actionable responses. It can understand a wide range of questions related to schools, such as school types (primary or secondary), performance metrics, location, and contact details.
* **Personalized Responses**:  
  Using NLP techniques, the chatbot processes user inputs and provides tailored responses. For instance, if a user asks for schools in a specific district or with certain performance levels, the chatbot delivers recommendations based on real-time data.
* **Speech Synthesis for Accessibility**:  
  The chatbot incorporates speech synthesis (text-to-speech) to ensure that the platform is accessible to users who prefer auditory interaction or those with limited literacy. This feature makes EduPortal more inclusive and user-friendly.
* **Seamless Integration with Recommendation System**:  
  The chatbot is tightly integrated with the recommendation system, powered by Graph Neural Networks (GNN) and collaborative filtering techniques. It utilizes the model’s output to provide users with accurate, relevant school suggestions, based on factors like geographic location, performance, and school phase.
* **Continuous Learning and Feedback Loop**:  
  As users interact with the chatbot, their feedback is collected and analyzed. This feedback helps refine the chatbot's performance, allowing it to better understand user preferences and provide more accurate recommendations over time.
* **Multi-Platform Accessibility**:  
  The chatbot is available across different platforms, including web, mobile, and possibly even through voice assistants. This ensures that users can easily access the service from various devices, enhancing its accessibility and overall usability.

**3. Achievability and Scalability**

The chatbot is built using robust, scalable technologies that ensure its functionality can grow with increasing user demand. By leveraging powerful NLP frameworks like **BERT** for understanding user queries and **Google’s Text-to-Speech** for generating speech responses, the chatbot is capable of handling large volumes of queries while maintaining accuracy and efficiency.

It also incorporates machine learning algorithms that continuously improve the chatbot’s ability to respond to new types of queries. As the system collects more data from user interactions, it can adapt and evolve to meet changing user needs, further enhancing its effectiveness.

In Conclusion, the Chatbot in EduPortal is not only relevant to the solution but also an essential tool for improving user experience. It is well-planned, appropriately integrated, and capable of delivering personalized, efficient responses to user queries. By leveraging advanced NLP and speech synthesis, along with a continuous learning mechanism, the chatbot ensures that EduPortal remains user-friendly, accurate, and accessible to a diverse set of users.

***Main.py:***

This script implements a Graph Neural Network (GNN)-based recommendation system using a Tkinter GUI.

The application allows users to generate school recommendations based on user reviews using a pre-trained model.

Key steps in the process:

1. Loading user and school data.

2. Preprocessing the data for model use.

3. Initializing and loading the pre-trained GNN model.

4. Making predictions using the model.

5. Evaluating the model's performance using various metrics (precision, recall, F1 score).

6. Generating 7and displaying recommendations for a sample user.

The user interface includes a button to trigger the recommendation process and a scrollable text box to display logs and results.

External Dependencies:

- gnn\_model: Contains the definition of the GNN model.

- data\_preparation: Functions to load and preprocess the data.

- evaluate\_and\_predict: Functions to evaluate the model's predictions.

- recommender: Function to generate recommendations based on a user embedding.

***Data\_preparation.py :***

This script loads and preprocesses user and school review data from CSV files.

Steps:

1. The 'load\_user\_data' function loads user reviews from a specified CSV file into a Pandas DataFrame.

2. The 'load\_school\_data' function loads school data from a specified CSV file into a Pandas DataFrame.

3. Missing values in both user and school data are handled by filling them with empty strings.

4. A sample of the user and school data is printed to provide an overview of the data structure.

5. If the file is not found, an error message is displayed, and the function returns None.

Dependencies:

- pandas for data manipulation

Parameters:

- 'generate\_user\_reviews\_with\_ids.csv' contains user review data.

- 'gauteng\_schools.csv' contains school information.

Returns:

- Preprocessed user and school data.

***recommendation logic.py :***

This file uses a trained Graph Neural Network (GNN) model to generate school recommendations based on user embeddings, which represent user preferences in numerical form. The `generate\_recommendations` function applies the GNN model to the input user embedding, using `torch.no\_grad()` to prevent the model from tracking gradients, as this is only an inference process. In the main script, the trained GNN model is loaded, initialized with specific input and output dimensions, and set to evaluation mode to ensure efficient performance. A sample user embedding is then provided to demonstrate the recommendation generation process, and the output recommendations are printed. This approach allows the GNN model to suggest schools tailored to individual user profiles.

***recommender.py :***

This function generates school recommendations for a given user based on the user's embedding and a trained model.

Steps:

1. The user embedding is passed to the model to generate recommendations.

2. The model output (recommendations) is returned.

3. torch.no\_grad() is used to ensure that no gradients are computed during the inference process, which saves memory and computation time.

Dependencies:

- torch for model inference and managing the user embedding.

Parameters:

- user\_embedding (torch.Tensor): The user embedding representing the user's features or preferences.

- model (torch.nn.Module): The trained model used to generate recommendations.

Returns:

- recommendations: The schools recommended by the model based on the user's embedding.

***train\_chatbot\_model.py:***

This script trains an enhanced neural network-based chatbot model to generate responses based on user questions.

Steps:

1. Data preprocessing includes loading the dataset, encoding labels, and vectorizing text using TF-IDF.

2. The data is split into training and validation sets.

3. A neural network model is defined with layers such as fully connected layers, batch normalization, dropout for regularization, and LeakyReLU activation.

4. The model is trained with a weighted loss function to handle class imbalance.

5. Early stopping is implemented to avoid overfitting, saving the best model based on validation loss.

6. The final model, along with the LabelEncoder and TF-IDF Vectorizer, are saved for future use.

Dependencies:

- torch for model training

- scikit-learn for data preprocessing and evaluation

- pickle for saving the vectorizer

Parameters:

- The dataset should contain 'question' and 'answer' columns, where 'question' is the input and 'answer' is the output.

- The model is trained with 50 epochs, early stopping with patience of 3, and a learning rate scheduler for adaptive learning rates.

Returns:

- Saved model ('chatbot\_model\_final.pth')

- Saved LabelEncoder classes ('label\_encoder\_classes.npy')

- Saved TF-IDF Vectorizer ('vectorizer.pkl')

***train\_gnn\_model.py:***

This script trains a Graph Neural Network (GNN) to predict school performance based on numeric features. It loads and preprocesses the data, splitting it into training and validation sets while standardizing the features. The script defines a GNN model with three layers, uses Cross-Entropy Loss, and optimizes with Adam. It trains the model for a set number of epochs, tracking loss and accuracy, then saves the trained model to a file. Finally, the script loads the saved model checkpoint to verify its parameters.

***gnn\_model.py:***

This script defines a simple Graph Neural Network (GNN) model using PyTorch.

The model is designed for tasks such as recommendation generation or classification.

Model Architecture:

1. A fully connected layer (fc1) transforms the input data to a hidden space with dimensionality `hidden\_dim`.

2. A second fully connected layer (fc2) further reduces the representation to a size of 32.

3. A final output layer (fc3) produces the logits (raw predictions) of size `output\_dim`.

Activation Functions:

- ReLU activation is applied after the first two fully connected layers to introduce non-linearity.

The model is suitable for tasks where the output is a set of logits or continuous values without activation.

Dependencies:

- PyTorch (torch) for defining and training the neural network model.

***generate\_user\_reviews\_with\_ids.py:***

This script generates random user reviews for a set of schools and saves the data to a CSV file.

Each school is assigned a random review and rating, simulating user feedback.

Steps:

1. A predefined set of school IDs and sample reviews are used.

2. For each school ID, a random review is selected and a random rating between 1 and 5 is generated.

3. The data (user ID, school ID, review, and rating) is stored in a pandas DataFrame.

4. The DataFrame is saved to a CSV file called 'generate\_user\_reviews\_with\_ids.csv'.

The generated data can be used for training recommendation models or other data analysis tasks.

Dependencies:

- pandas for data manipulation and file saving.

- random for generating random reviews and ratings.

***gauteng\_schools.py:***

This code creates a DataFrame from a dictionary containing information about various schools in Gauteng. The data includes columns for school ID, school name, school phase (Primary or Secondary), the address of the school, the education district, the performance percentage for 2023, and contact details (email and phone). Once the DataFrame is populated with the information, it is saved as a CSV file named 'gauteng\_schools.csv'. The CSV file is generated without the index, providing a structured and accessible format of the school data that can be further analyzed or utilized.

***evaluate\_and\_predict.py:***

This Python script evaluates the performance of a Graph Neural Network (GNN) model for school recommendation tasks. It first loads user review data and school data from CSV files. The script then processes the user data through the trained GNN model, generating predictions for each data point. For each prediction, the model outputs the top 7 predicted school IDs. These predictions are compared to the true school IDs (from the `user\_data`), and several evaluation metrics—precision, recall, F1 score, and Mean Average Precision (MAP)—are calculated to assess the model's accuracy and effectiveness. The evaluation results are printed out, giving a clear view of how well the model performs in terms of these key metrics. The script also uses a random validation set generated based on the user data for testing the model.

***chatbot\_dataset.py:***

This Python script is designed to expand a chatbot's training dataset by generating a larger set of question-answer pairs. It starts by defining a set of additional questions and corresponding answers related to school recommendations. Then, it multiplies these pairs to simulate a larger dataset, further modifying the questions by introducing variations using a predefined list of terms. This process ensures greater diversity in the dataset, which can improve the chatbot's ability to understand and respond to a wider range of queries. Finally, the expanded dataset is saved to a CSV file for use in training the chatbot.

***chat\_bot.py:***

This Python script implements a chatbot that uses a deep learning model built with PyTorch for intent classification. It first vectorizes the user input using a pre-trained TF-IDF vectorizer and then passes the vectorized input through a neural network to make predictions. The output from the model is decoded using a LabelEncoder, which maps the predicted label to a meaningful response. The chatbot integrates text-to-speech (TTS) functionality via the `pyttsx3` library, allowing it to speak its responses aloud. Users interact with the chatbot by typing their input in the terminal, and the chatbot responds both in text and via voice.