Design Doc for Intervention Prediction AI Tool - Team3

1. Overview

The Beam Group company is an Ontario-based company that provides a full range of services to help organizations address strategic, operational, and public policy challenges. The company brings together a broad set of skills and experience including research, management consulting, and public policy development to deliver comprehensive client-focused solutions.

The goal of the project is to build a prototype of an AI-based tool that public sector organizations of the company can use to predict service or intervention needs based on client characteristics and results.

By following the designs below, we aim to create an AI model that not only meets the current requirements but is also scalable and adaptable to future needs, ensuring long-term value for the Beam Group and its public sector clients.

2. Requirements

- An AI model is to be developed to determine the most effective intervention for clients based on user-provided information.
- This model will process a dataset containing historical client data, including characteristics and the interventions applied to them.
- The model should be capable of analyzing this dataset and performing classification analysis.
- Upon development and training, the model must allow Case Management workers to input specific client information and receive corresponding analysis results.

2.1 User Stories

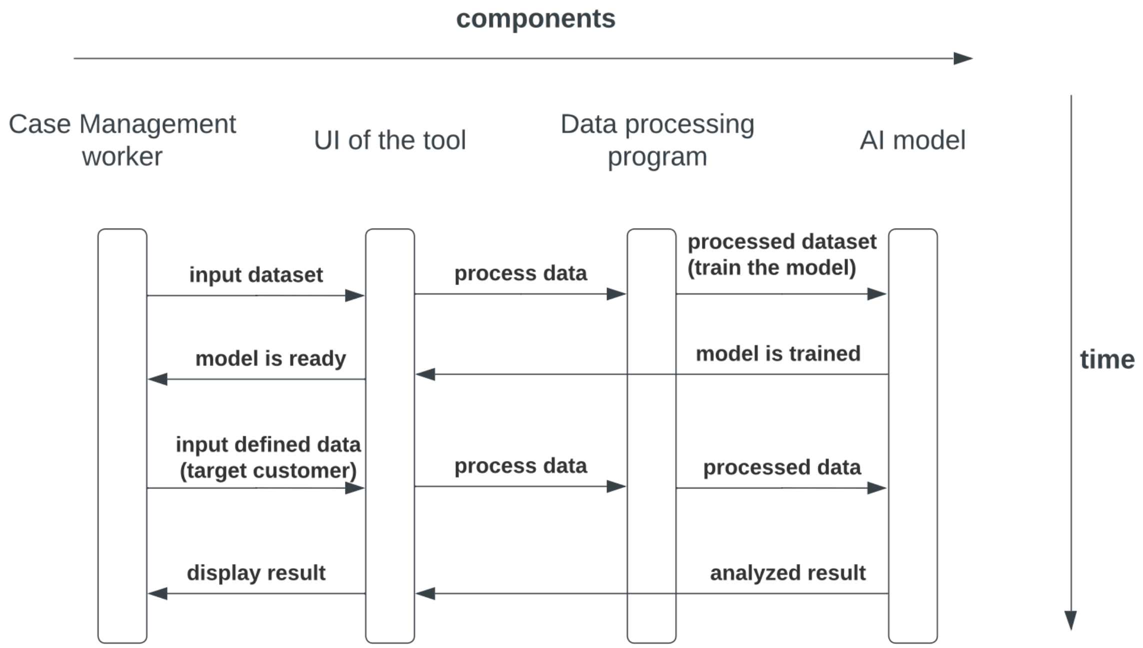
As a Case Management worker, I want to upload a CSV file containing a client's characteristic information to the AI tool. The tool should then determine the most suitable intervention based on this data and display the results on a dashboard.

- When a user uploads a CSV dataset file and initiates the analysis, the AI tool should extract and analyze the data from the file, and generate a classification model from the given dataset.
- When a user uploads a CSV file for a target customer, the AI tool should extract and analyze the data from the file, and identify the most appropriate intervention. The results should be visible on the dashboard.

- An optional feature would display all potential interventions on the dashboard, ranked from the most to the least beneficial based on the model's analysis, and grouped by intervention categories.
- The tool should gracefully handle cases where the CSV file is empty contains no data, or some data is empty, ensuring robust file processing.
- Initially, the tool will support only CSV file formats, considering the client's current data management practices. Support for additional file types may be considered for future tool enhancements.

3. High-Level Design

3.1 Overview of the design



According to the UML sequence diagram above, the user is required to input a dataset via the tool's user interface. This dataset should then be forwarded to a data processing program, where it will undergo cleaning before being inserted into the database. Subsequently, the AI model will be trained using the cleansed data from the database. Upon successful training, a notification will be dispatched to the user, indicating that the model is ready for use.

Following receipt of this notification, the user can input specific data—namely, information about the target customer—through the user interface. This data will be processed by the program and forwarded to the AI model for analysis. The resulting analysis will then be relayed back to the user interface and displayed to the user in an understandable format.

3.2 Database

For better data management purposes, we should create a database to store our data instead of using the CSV file directly.

The benefits of using a database:

- **Data Integrity and Relationships:** Databases enforce rules regarding data integrity and relationships between datasets, ensuring data consistency and accuracy. Constraints and validation rules can prevent invalid data entry.
- **Scalability:** Databases are designed to handle large volumes of data and can be scaled up or distributed across multiple servers as needed, unlike CSV files which can become unwieldy as they grow in size.
- **Backup and Recovery:** Most database systems have built-in support for data backup and recovery, ensuring data can be restored in case of corruption or loss. Managing backups with CSV files would be a manual and error-prone process.
- **Supportive Tools:** Databases are supported by a wide range of management, development, and analytical tools that make it easier to work with data, including automated schema changes, indexing, and performance tuning.

Disadvantages of using a database:

- **Complexity:** Setting up and maintaining a database requires a certain level of expertise in database administration, which can be more complex than managing simple CSV files.
- **Portability:** CSV files are highly portable and can be easily shared, opened, and edited with a wide range of software, from simple text editors to complex data analysis tools. Databases, on the other hand, may require specific drivers, tools, or exports to access data.

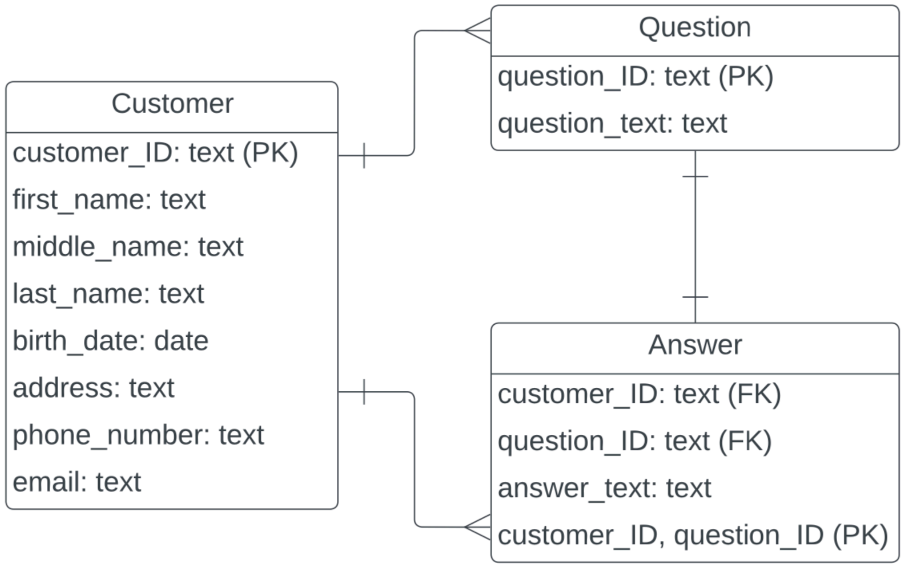
Conclusion: Opting for a database to store data offers a more organized, professional, and maintainable approach for future development, despite some complexities and requirements.

3.2.1 Database options

- **Relational database**: Given that our data is highly structured and fits neatly into tables, it is advisable to use a relational database to store the information currently held in the CSV file.
- **NoSQL database**: Although NoSQL databases are easily scalable for large datasets, our current project is in the prototype phase, and scalability is not our primary concern. Therefore, we will not consider a NoSQL database for this project.

Conclusion: We will use a relational database to manage our data. Suitable options include MySQL and SQLite, both of which are well-suited to our requirements at this stage.

3.2.2 Data Schema



According to the UML class diagram above, the schema of the database should be as follows:

```
Customer (
   customer_ID : text (Primary Key),
   first_name : text,
   middle_name : text,
   last_name : text,
   birth_date : date,
   address : text,
   phone_number : text,
   email : text
)

Question (
   question_ID : text (Primary Key),
   question_text : text
)

Answer (
   customer_ID : text (Foreign Key references Customer(customer_ID)),
   question_ID : text (Foreign Key references Question(question_ID)),
   answer_text : text,
   PRIMARY KEY (customer_ID, question_ID)
)
```

3.3 Data Processing Program

3.3.1 Requirements

- The data from the dataset must be cleaned before being inserted into the database.
- The program must accurately read the data from the CSV file.
- The program must effectively process the data by either removing or assigning values to empty fields, according to the user's specifications.
- The program must correctly insert the cleaned data into the database.
- The program must precisely process the target customer's data, based on user input, and forward it to the AI model.

3.3.2 High-Level Design

Combining Python with a MySQL database is a popular and effective approach for a program that reads data from a CSV file, processes it, and then inserts it into the database. Utilizing Python's robust libraries, such as “***pandas***” for data manipulation and the “***csv***” module for CSV file handling, simplifies the tasks of reading, processing, and cleaning data. Additionally, the “***mysql-connector-python***” library facilitates an efficient connection to MySQL databases, ensuring seamless data insertion.

3.4 AI Model

3.4.1 Requirements

- The model must utilize classification techniques to analyze the data stored in the database.
- It should be capable of ingesting new data and accurately classifying it based on the trained model.

3.4.2 High-Level Design

The AI model will be designed to integrate seamlessly with the data processing program and database, forming a comprehensive system that enables the Beam Group to predict service or intervention needs for public sector organizations.

- **Model Selection**: Based on the requirement for classification, we will evaluate various machine learning algorithms, such as Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines (SVM), to select the most appropriate model that balances accuracy and computational efficiency.
- **Data Preprocessing**: Before training, the data will undergo preprocessing to ensure it's in the optimal format for the model. This includes normalizing or standardizing numerical data, encoding categorical variables, handling missing values, and potentially reducing dimensionality to improve model performance.

- **Feature Engineering**: We will extract and select relevant features from the cleansed data that significantly impact the model's ability to accurately classify client needs. This might involve analyzing historical intervention outcomes to identify the most predictive factors.
- **Model Training and Validation**: The selected model will be trained using some of the cleansed and processed data. We will employ cross-validation techniques to gauge the model's performance accurately and prevent overfitting, ensuring that the model generalizes well to unseen data.
- **Integration with Data Processing Program**: The trained model will be integrated with the existing data processing program. This integration allows the model to receive processed client data as input, perform classification, and then output the predicted intervention needs.
- **User Interface Feedback**: Once the model makes a prediction, the result will be sent back to the user interface, where it will be displayed to Case Management workers. The interface will clearly present the recommended intervention, along with a confidence score or probability to aid decision-making.
- **Continuous Learning**: To maintain and improve the model's accuracy over time, a mechanism for retraining the model with new data will be established. This may involve periodically updating the model with additional data collected from recent client interactions and outcomes.

4. Low-Level Design

5. Timeline Schedule

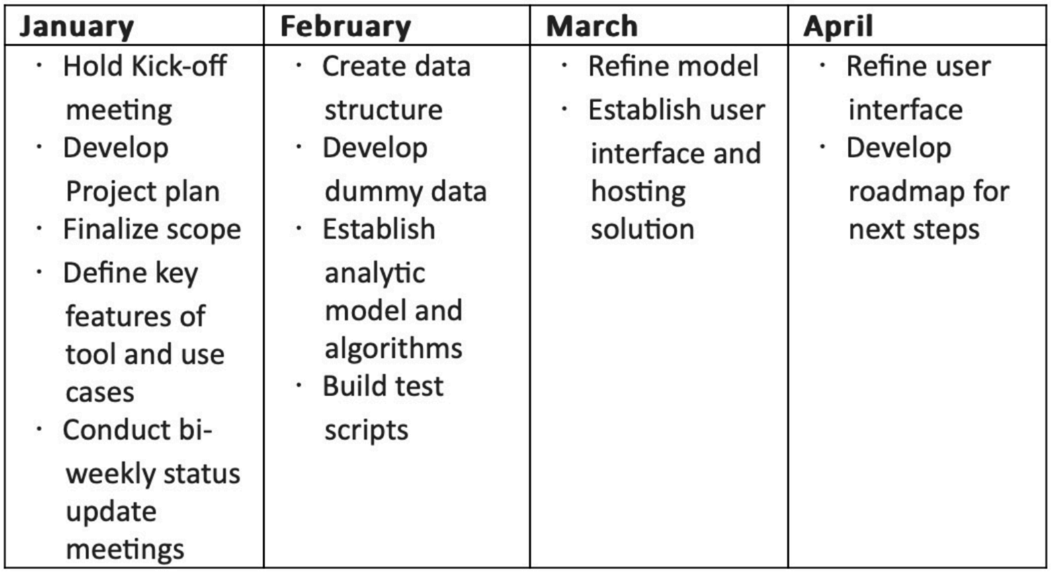
	Timeline	Tasks Need To Be Done	Status	Notes
Sprint1	Week1 1/22-1/26	Gathering requirements from clients	Completed	
		Understanding the requirements	Completed	
	Week2 1/29-2/2	Create user stories	Completed	
		Create UML diagrams	Completed	
		Create example dummy data	Completed	
Sprint2	Week3	Confirm the requirements with clients	In Progress	

	2/5-2/9	Continue to complete the design doc	In Progress	
			Not Started	
	Week4 2/12-2/16		Not Started	
			Not Started	
			Not Started	
Sprint3	Week5 2/19-2/23		Not Started	
			Not Started	
			Not Started	
	Week6 2/26-3/1		Not Started	
			Not Started	
			Not Started	
Sprint4	Week7 3/4-3/8		Not Started	
			Not Started	
			Not Started	
	Week8 3/11-3/15		Not Started	
			Not Started	
			Not Started	
Sprint5	Week9 3/18-3/22		Not Started	
			Not Started	
			Not Started	
	Week10 3/25-3/29		Not Started	
			Not Started	
			Not Started	

Week 11 finalize everything

Memo(Will be deleted)

High Level Timeline



January	February	March	April
• Hold Kick-off meeting • Develop Project plan • Finalize scope • Define key features of tool and use cases • Conduct bi-weekly status update meetings	• Create data structure • Develop dummy data • Establish analytic model and algorithms • Build test scripts	• Refine model • Establish user interface and hosting solution	• Refine user interface • Develop roadmap for next steps

What are the requirements? Are we going to get a Doc?

We don't know yet. No, we need to generate the Doc by ourselves. We need to write down the requirements from the meeting.

Is the kickoff going to be recorded? Review the info from the customer.

Yes, it will be recorded.

What does the data look like?

Not sure yet. It depends on how we want to train our model. We could mine the data, do some data processing, and then feed it to our model.

Do we need to build a user interface? GUI?

It will be nice to have one, but an executable file will be okay too.

Do we need to set up a database for training and testing?

It will be good to have a database since it will be easier to manage the data through a database when the data becomes big. Some tools could be MySQL/SQLite.

After the design, is the customer/TA/Instructor going to review the design doc? Make sure the project is on the right track.

The customer should have access to the Doc, but the TA will definitely go through the Doc and review it for us.

Meeting 1 Kickoff:
Classification model, binary result
Give caseworkers a data-driven tool to determine what interventions should be taken
Would the client succeed return to work due to intervention ABC
Recommendation tool for case managers

Data will come in with a CSV/JSON/XML file
The more important part is the predicted power
No real data will be provided
A dashboard to show the result
The tasks for the next sprint: generate dummy data to be fed into the system, structure of the data

Summary:
A data-driven tool is planned to be developed for the user for job-search or related interventions. The goal is to provide the best interventions and services to the client. The outcome of the tool —like the salaries of a worker — is desired to know. In the future want to find out the relationship between intervention and outcome from actual real data.