Early Disease Prediction

Section A1 - Team 1

Rebecca Bubis, Sanjal Desai, Yashna Meher, Mishil Trivedi

3 March 2025

**Proposal**

Early warning signs of diseases often go unnoticed due to the lack of structured screening processes, leading healthcare professionals to rely on subjective assessments. Traditional diagnostic methods, while effective, are often labor-intensive and may fail to detect subtle patterns in patient health data. This gap highlights the need for a data-driven approach to improve early disease detection and intervention.

This project aims to leverage unsupervised machine learning to analyze medical records based on symptoms and health conditions, with a focus on identifying disease risk profiles based on biochemical patterns to enhance early diagnosis and prevention strategies. By clustering patients exhibiting early-stage disease patterns, our approach seeks to uncover hidden correlations between demographic, lifestyle, and clinical factors. This will enable healthcare providers to identify high-risk individuals earlier, facilitating timely interventions and a more proactive approach to disease prevention. Ultimately, this project aims to enhance diagnostic accuracy, optimize early intervention strategies, and improve overall patient outcomes through data-driven insights.

This project presents several key challenges that must be carefully navigated to ensure a robust and meaningful analysis. The first challenge lies in selecting the appropriate files from the extensive NHANES dataset. Given the vast scope of data available, identifying the relevant files while avoiding redundancy is critical. Once selected, the next hurdle is efficiently merging these files to create a cohesive dataset for analysis.Another major challenge is handling missing values. Given the dataset’s size and complexity, determining an acceptable threshold for missing data in each column is essential. Striking a balance between retaining valuable information and preserving data integrity requires thoughtful decision-making. Feature selection poses yet another challenge. With a high-dimensional dataset, reducing the number of variables while ensuring that key relationships—both linear and nonlinear—are captured is crucial for meaningful insights.

Additionally, since this is a medical dataset, the presence of outliers is expected. Developing a model that can effectively handle these outliers without compromising accuracy or generalizability adds another layer of complexity. Addressing these challenges strategically will be key to deriving accurate and insightful conclusions from the data.

**Preliminary Preprocessing**

The dataset that we have, required us to join close to 35 files which were spread out amongst the Demographic Data, Dietary Data, Examination Data, Laboratory Data, and Questionnaire Data files. Some columns described the person’s smoking history, with a yes or no and also their eating habits. They also highlighted their fitness history, all of which can be a factor in prediction of the said disease. There are also columns that highlight a person’s diabetes history which is another important factor in early disease prediction. Each file contained very valuable data related to each person’s physical health and their health history. Initially, it was decided to eliminate files that had around 70% of the data as missing values. When that eliminated a small portion of the columns of the combined data, we decided to be more bold and removed all columns that had 50% missing data.After that, we went ahead and dropped all the missing values. In a medical dataset, assuming a value to be “0” holds certain significance. For instance, a dataset tracking diseases, 0 might indicate the absence of a condition (e.g., 0 = No Diabetes, 1 = Diabetes). Misinterpreting 0 as missing data instead of a valid category could lead to incorrect conclusions. Hence, we decided to go ahead with this aggressive approach to clean the data. Given the complexity and size of the dataset—spanning over 400 columns across multiple domains—significant effort was first dedicated to cleaning the data, handling missing values, and reducing dimensionality.

**Exploratory Data Analysis (EDA) and Preprocessing**

Due to the extensive size of our dataset, conducting exploratory data analysis (EDA) presents certain challenges, particularly in deriving meaningful insights from summary statistics and visualization techniques. The dataset integrates Demographic Data, Dietary Data, Examination Data, Laboratory Data, and Questionnaire Data, resulting in a combined structure of over 400 columns which was reduced to the dataset containing 11,933 rows and 226 columns. Given this scale, traditional summary statistics, heatmaps, and pairplots offer limited interpretability.

To handle missing data, we applied a strict filtering approach, dropping columns where more than 50% of values were missing and removing incomplete rows to prevent biases in clustering. This step reduced the dataset while maintaining its reliability. Additionally, duplicate patient entries were identified and removed, bringing the total down to 1,273 unique records. To understand relationships between different health attributes, we conducted correlation analysis and found that while most features were weakly related, some exhibited strong connections, particularly among laboratory test results and blood pressure measurements. However, notable correlation clusters appeared in the upper-left and bottom-right sections. Despite these localized patterns, the overall lack of significant linear relationships suggests that Principal Component Analysis (PCA) may not be the most effective dimensionality reduction technique. Nevertheless, we will continue to explore PCA while also considering alternative nonlinear approaches.

Instead of immediately removing redundant features, we used network graphs to visualize these relationships and better understand the structure of the data. These graphs revealed distinct clusters of highly correlated attributes, suggesting that some variables carried overlapping information. However, rather than eliminating these features, we retained all of them for clustering, ensuring that we didn’t discard any potentially important medical data.

Regarding data preprocessing, we have undertaken several key steps to ensure the integrity and usability of the dataset. Given the diversity of variables, we applied standardization to bring all numerical features to a common scale. This step ensured that no single measurement, such as weight or cholesterol levels, dominated the clustering process. Additionally, we examined outliers using reconstruction errors, ensuring that extreme values were carefully reviewed rather than arbitrarily removed. Since medical data often contains natural variability, we took a conservative approach to outlier handling, keeping potentially valuable patient data in the analysis.

With the dataset now cleaned and standardized, we moved forward with hierarchical clustering on the full numerical dataset.

**Analysis & Experiments**

Understanding the Clustering Methods Used in the Project

In this project, various clustering methods were explored to group patients based on their symptom patterns and health indicators. The primary goal was to uncover hidden structures in the data that could be useful for early disease detection. Each method was chosen for its strengths in handling different aspects of the dataset, such as high dimensionality, nonlinear relationships, and varying cluster shapes. Below is a detailed explanation of each clustering technique and the reasoning behind its use.

1. Hierarchical Clustering

Hierarchical clustering was the first method applied to explore how features and patients naturally group together. Instead of specifying the number of clusters beforehand, this approach builds a hierarchy of clusters based on feature similarity. The Ward linkage method was used, which minimizes the variance within each cluster as the hierarchy is built. The algorithm generated a dendrogram, a tree-like visualization, to show how features are related based on their distances in the correlation matrix.

This method was particularly useful for identifying relationships between features before moving on to clustering patients. By applying hierarchical clustering to the feature set, we could observe which variables were strongly connected and could potentially be grouped together. The main advantage of this approach is its interpretability—rather than just assigning a label to each patient, we could visually inspect how attributes were grouped and determine if meaningful health-related clusters emerged.

2. Principal Component Analysis (PCA) for Dimensionality Reduction

Although initial correlation analysis suggested weak linear relationships, PCA was tested to see if it could capture important variance in the dataset while reducing dimensionality. The idea was to transform the high-dimensional dataset into a smaller set of principal components that retain most of the information.

By analyzing explained variance, we determined that 85 principal components were needed to retain 90% of the original variance. This helped compress the dataset while preserving meaningful patterns. PCA was particularly useful for removing noise and ensuring that redundant features did not skew the clustering results. After dimensionality reduction, we applied clustering algorithms like K-Means to the transformed dataset.

3. K-Means Clustering

K-Means is a widely used clustering algorithm due to its simplicity and efficiency. The goal was to group patients into distinct clusters based on their transformed features. We first determined the optimal number of clusters using the Elbow Method, which suggested k = 5 as a good balance between cluster separation and compactness.

Once the clusters were formed, the next step was to interpret the characteristics of each group. By analyzing how each principal component contributed to the clustering, we aimed to understand what distinguished one group of patients from another. This was a key step in identifying whether certain clusters corresponded to early indicators of disease.

While K-Means works well with structured and well-separated clusters, it has limitations when dealing with overlapping or irregularly shaped clusters, which led us to explore alternative methods.

4. UMAP for Nonlinear Dimensionality Reduction

Since PCA is a linear technique, we also tested Uniform Manifold Approximation and Projection (UMAP) to capture nonlinear structures in the dataset. UMAP is particularly useful when relationships in the data are complex and not well-represented in a straight-line fashion.

By applying UMAP, we aimed to better visualize and understand the underlying patient clusters before applying K-Means. Unlike PCA, which maintains global variance, UMAP focuses on local structures, meaning it preserves the relationships between patients with similar symptoms. The reduced dataset from UMAP was then used for clustering analysis, allowing us to compare its effectiveness against PCA-based clustering.

5. Outlier Detection Using PCA Reconstruction Errors

Medical datasets often contain outliers, some of which represent true medical conditions rather than data errors. Instead of removing extreme values arbitrarily, we used PCA reconstruction errors to identify potential outliers. The idea was to reconstruct the original dataset from its PCA-transformed version and measure how much information was lost.

Patients with the highest reconstruction errors were flagged as potential outliers. This step was crucial in ensuring that genuine medical conditions were not mistakenly discarded while filtering out erroneous or inconsistent data points.

6. CUR matrix decomposition

In this project, CUR matrix decomposition was explored as an alternative dimensionality reduction technique to retain interpretability while reducing dataset complexity. Unlike PCA, which transforms data into principal components, CUR selects actual columns from the dataset, preserving their original medical meaning. This approach was particularly useful in ensuring that important health-related variables remained identifiable after dimensionality reduction. By applying CUR, we aimed to reduce the number of features while maintaining key medical insights, allowing us to perform clustering on a more manageable yet interpretable subset of attributes. However, our subsequent analysis revealed significant limitations in the clustering techniques applied. Hierarchical clustering, while effective in identifying relationships between data points, lacked predictive power since it does not generalize to new cases or learn from labeled disease outcomes. The 3D hierarchical clustering plot further illustrated this limitation by visualizing clusters without offering a probabilistic assessment or classification framework. Similarly, K-Means clustering with t-SNE provided a useful visualization of patient subgroups but failed to generate a model capable of predicting disease likelihood in new patients. Even DBSCAN, evaluated using the elbow method for selecting the ε parameter, proved ineffective, as the plot lacked a clear elbow point, making it difficult to determine an optimal clustering threshold.

Why So Many Methods?

Each clustering method was chosen to complement the others and help us analyze the dataset from multiple perspectives:

* Hierarchical Clustering helped explore feature relationships before clustering patients.
* PCA provided a way to reduce dimensionality while preserving most of the data’s information.
* K-Means was used for grouping patients, leveraging the compact representation provided by PCA.
* UMAP offered a nonlinear alternative to PCA, potentially revealing hidden patterns missed by linear techniques.
* Outlier detection ensured that data inconsistencies didn’t distort the clustering process.

By combining these methods, we could compare different approaches and determine which clustering strategy provided the most meaningful insights for early disease detection.

**Challenges, Dead Ends, & Adjustments**

One of the challenges our team faced was implementing PCA in a dataset that lacked linearity. As stated before, we suspected that it might be an ineffective method at dimensionality reduction, but we wanted to try it as a baseline approach since we found some attributes that exhibited some correlations during EDA. This approach ultimately failed, leaving the team with over 80 columns to retain 90% of the variance. Attempting to visualize clusters with so much information would be impractical and meaningless as we wouldn’t be able to draw interpretable conclusions from it. Moving forward, we chose to look at nonlinear methods for dimensionality reduction in the hopes that it would be more effective in identifying the most important columns.

The challenges with CUR became evident as its probabilistic column selection led to the omission of potentially important medical variables, limiting the dataset’s representational power. This lack of explicit feature optimization meant that key health predictors could be overlooked, making it unsuitable for precise medical analysis. To compensate, dimensionality reduction techniques like t-SNE and PCA-reduced K-means were attempted, but both resulted in unclear cluster distinctions, making interpretation difficult. Further, DBSCAN struggled due to varying density in the dataset, failing to form meaningful clusters, while hierarchical clustering, despite achieving a high silhouette score, only produced three broad clusters with low visual interpretability, ultimately proving ineffective for extracting actionable insights.

**Findings and Interpretations**

**Method 1**

Looking at the cluster output above, notice that the clusters represent the components and not the original column names. We identified the original columns with the highest loadings for each of the components below:

* WTPH2YR\_y is the Two-Year Examination Weight
* BMXWT is Body Weight (kg)
* LBXTST is Total Serum Testosterone
* LBDESTSI is Estimated Insulin Sensitivity Index (ISI)
* LBDFOT is Folate (Vitamin B9) Level in Blood

Cluster 0 (blue) consists of individuals with moderate weight, normal testosterone, and moderate insulin sensitivity, suggesting this group may have lower risk of liver disease and stroke. Cluster 1 (orange) looks to be a higher-risk group for stroke characterized by higher weight, lower insulin sensitivity, and lower testosterone. Cluster 2 (purple) likely represents a younger and healthier group with normal insulin, sensitivity, weight, and vitamin B9 levels. Cluster 3 (pink) includes individuals at higher risk for stroke and liver dysfunction shown through low folate, high weight, and insulin resistance. Finally, Cluster 4 (green) appears to be a healthier group with higher folate, improved insulin sensitivity, and moderate weight.

**Method 2**

The hierarchical clustering dendrogram, the 3D hierarchical clustering plot, and the K-Means clustering with t-SNE all proved unsuitable for the goal of early disease prediction. Hierarchical clustering is useful for identifying relationships between data points but lacks predictive power since it does not learn from labeled disease outcomes or generalize to new cases. The 3D hierarchical clustering plot further illustrates this limitation by showing clusters without any probabilistic assessment or classification framework. Similarly, K-Means clustering with t-SNE helps visualize patient subgroups but does not provide a model that can predict disease likelihood in new patients. Even the elbow method plot for selecting the DBSCAN epsilon (ε) parameter shows a poor scenario for unsupervised learning, as it lacks a well-defined elbow point. Instead of a clear transition, the plot exhibits a gradual increase followed by a sharp rise at the end, making it difficult to determine an optimal ε value.

### 

### **Method 3 ( Best Results)**

### UMAP effectively reduces the dimensionality of high-dimensional patient data, revealing hidden structures. However, while it helps visualize patient groups, it does not inherently predict disease outcomes. The 2D UMAP projection provides an intuitive representation of how patients are clustered, but without a supervised learning approach, it lacks predictive capability. Even when combined with clustering methods like K-Means or DBSCAN, the approach does not generalize well to unseen data, limiting its usefulness for early disease prediction. As it has given following outputs it was the best method for us in the given scale and format.

Cluster 0 - Cardiovascular Disease, Heavy Metal Toxicity, Metabolic Syndrome

Cluster 1 - Hypertension, Cardiovascular Disease, Metabolic Syndrome

Cluster 2 - Hypertension, Blood Disorders, Parathyroid Disorders (Osteoporosis)

Cluster 3 - Hormonal Disorders (PCOS, Hypogonadism), Cardiovascular Disease, Kidney Dysfunction

**Appendix**

**Contribution**:

Rebecca Bubis: Pre-processing, preliminary EDA, and PCA with KMeans clustering.   
Sanjal Desai: Completed Umap approach with different clustering methods, compared the methods used, tried permutations for CUR Approach, Compilation/version control of notebook

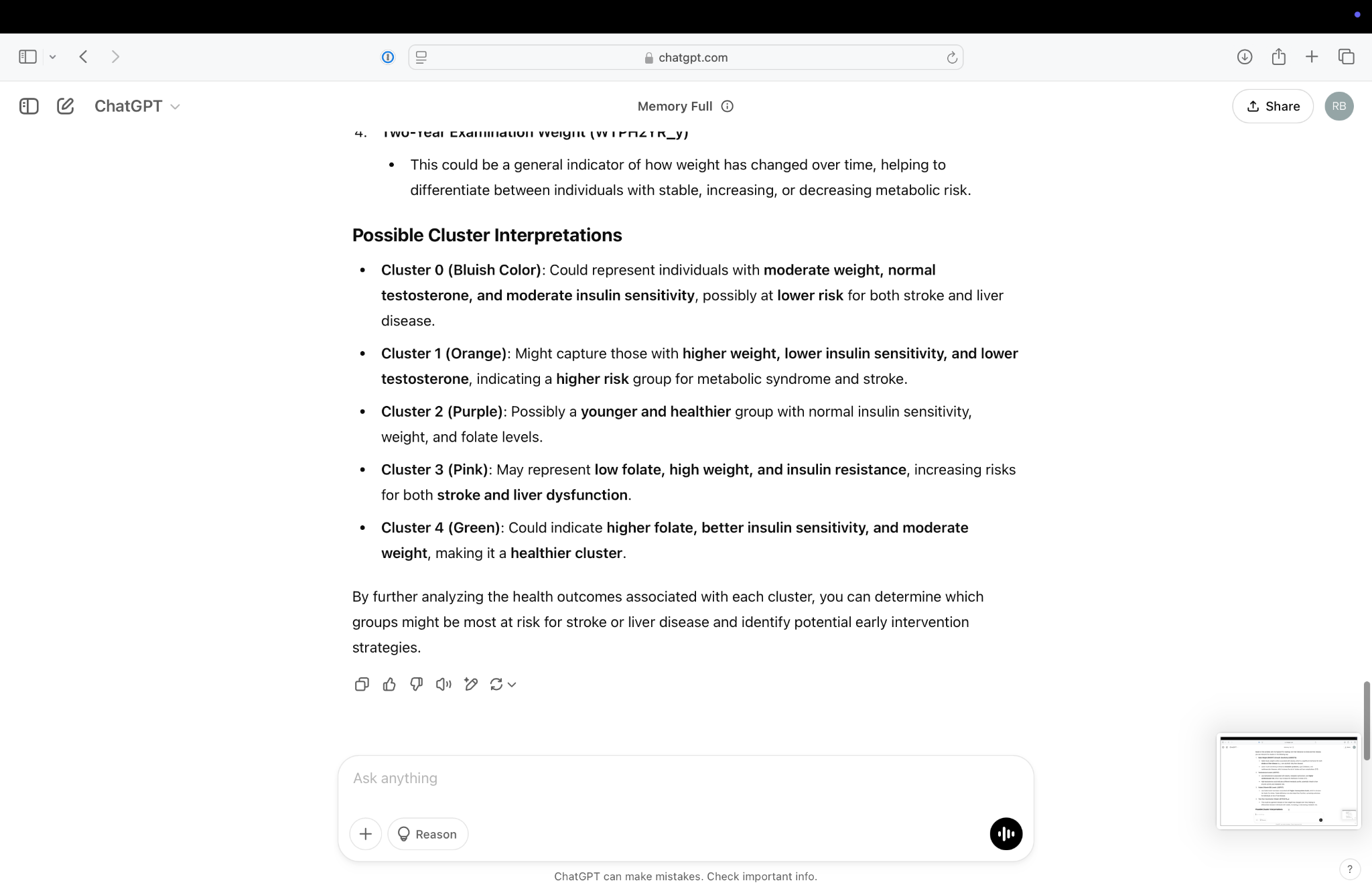
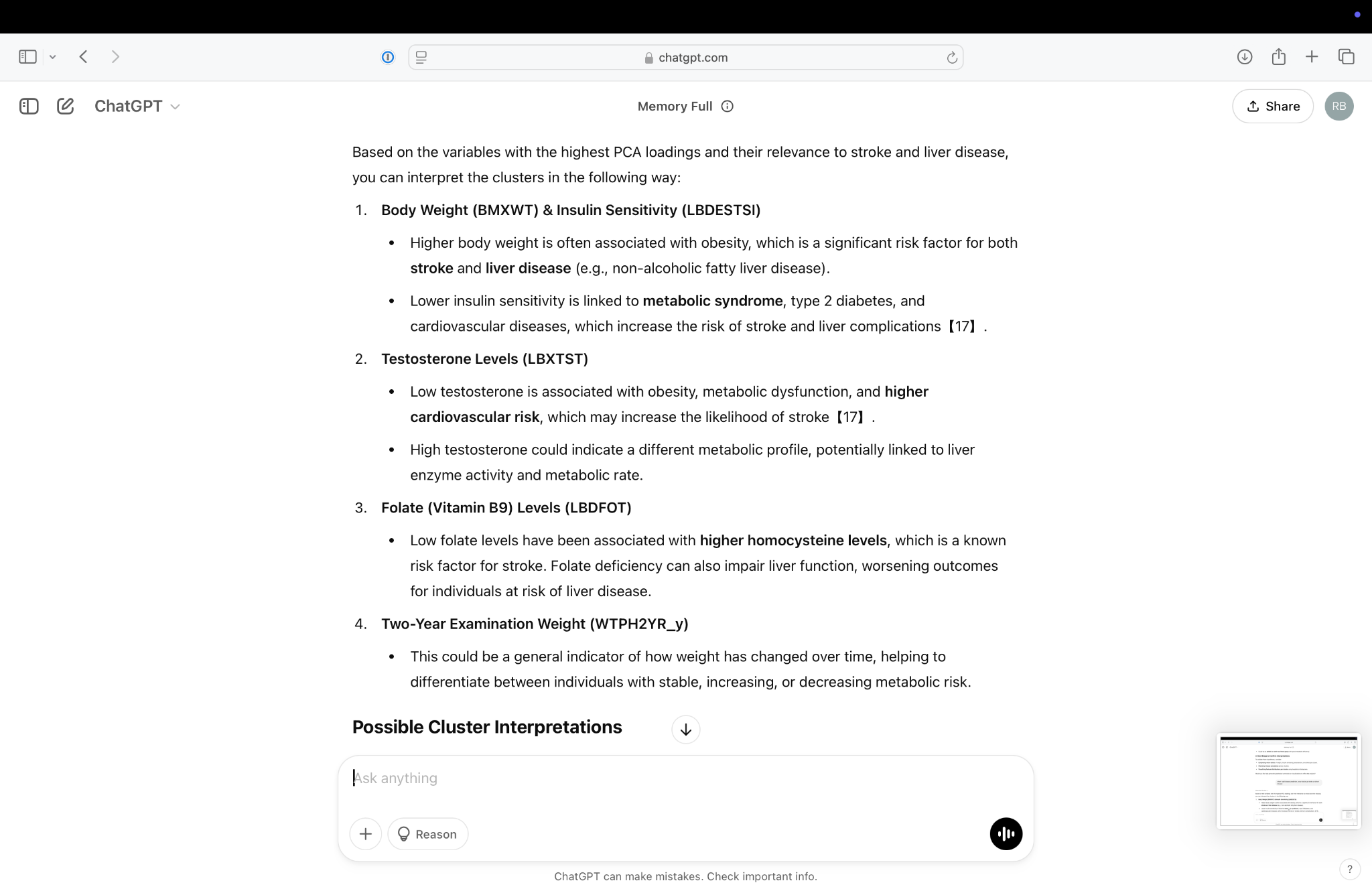
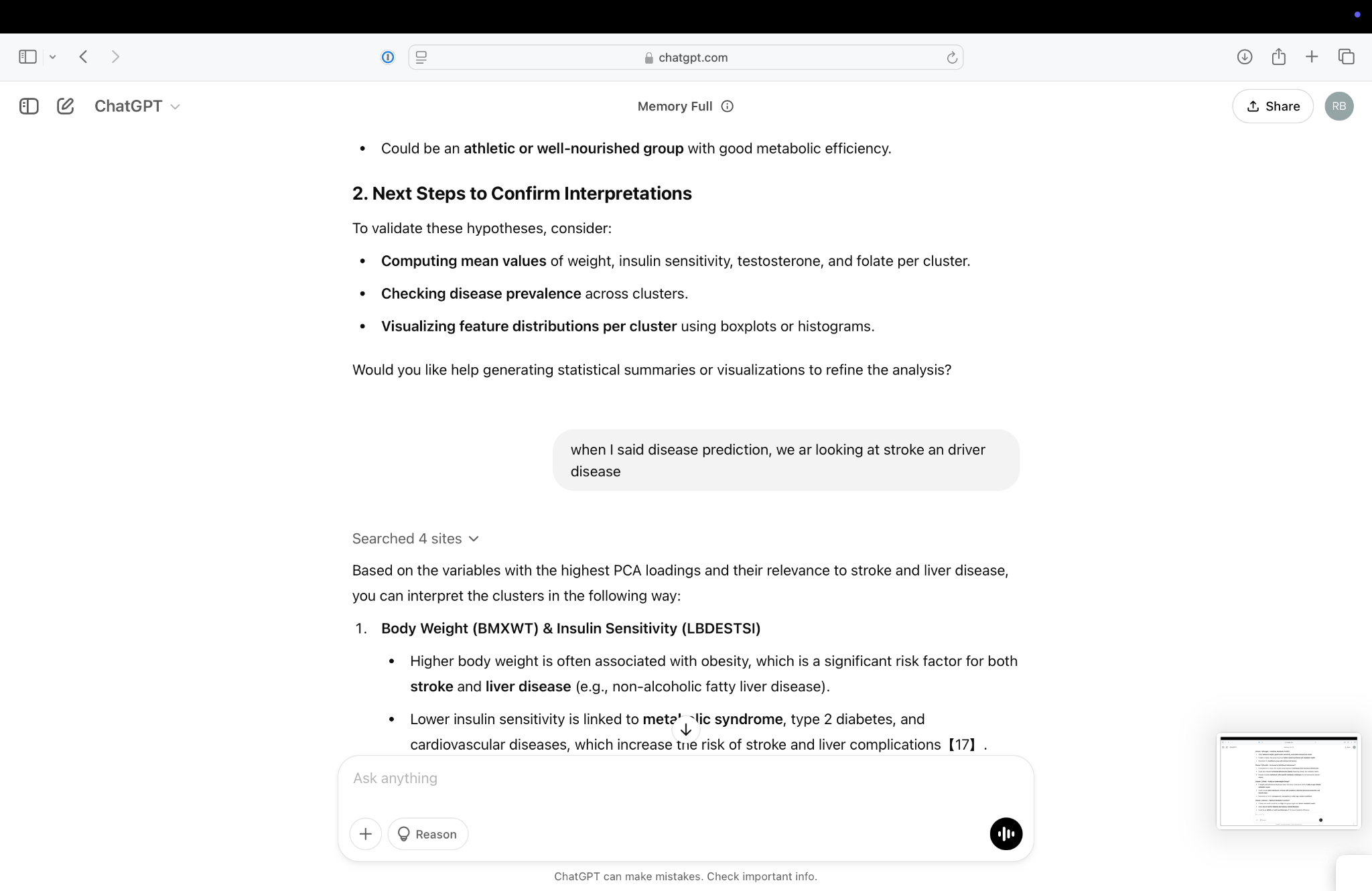
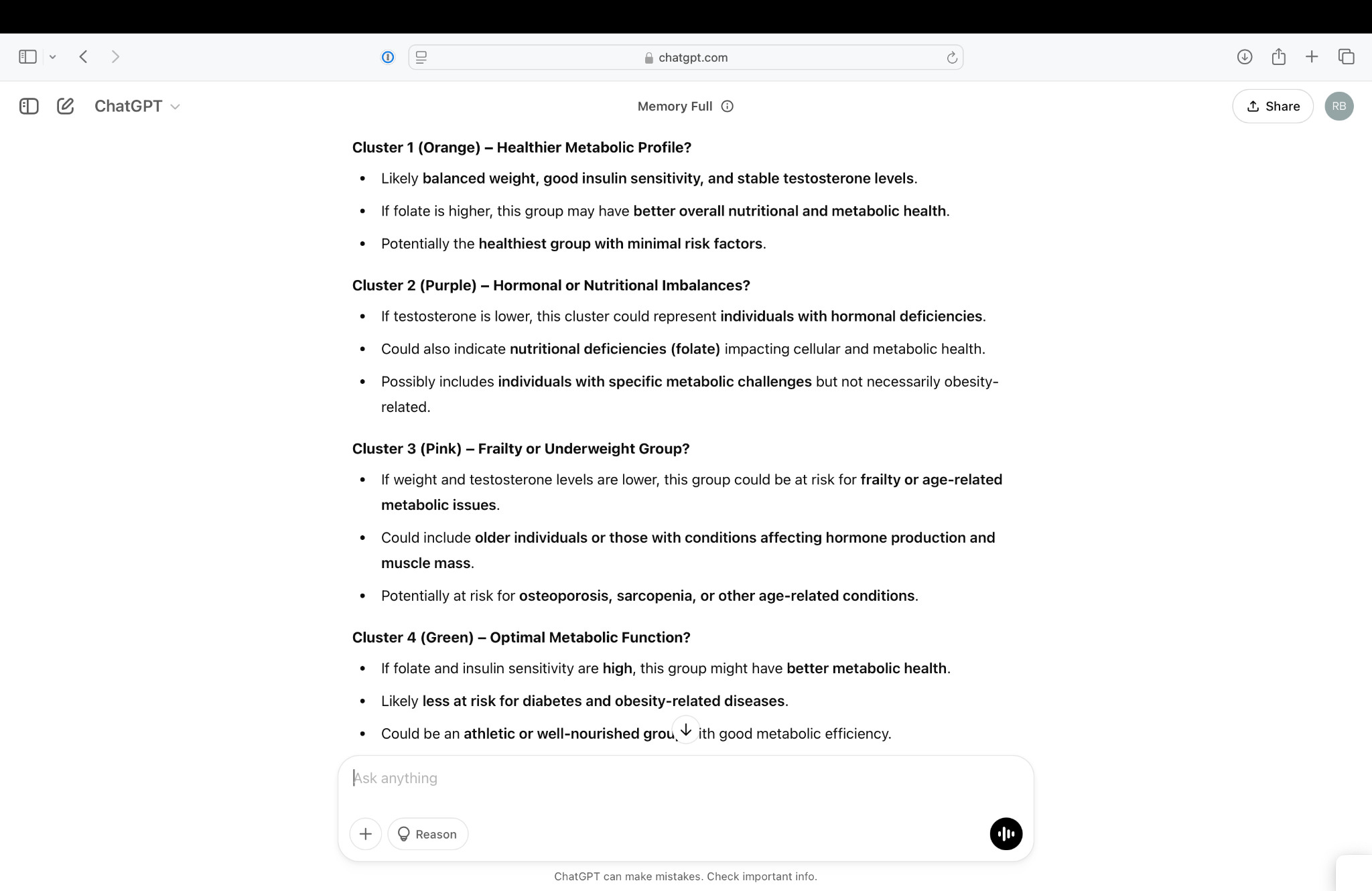
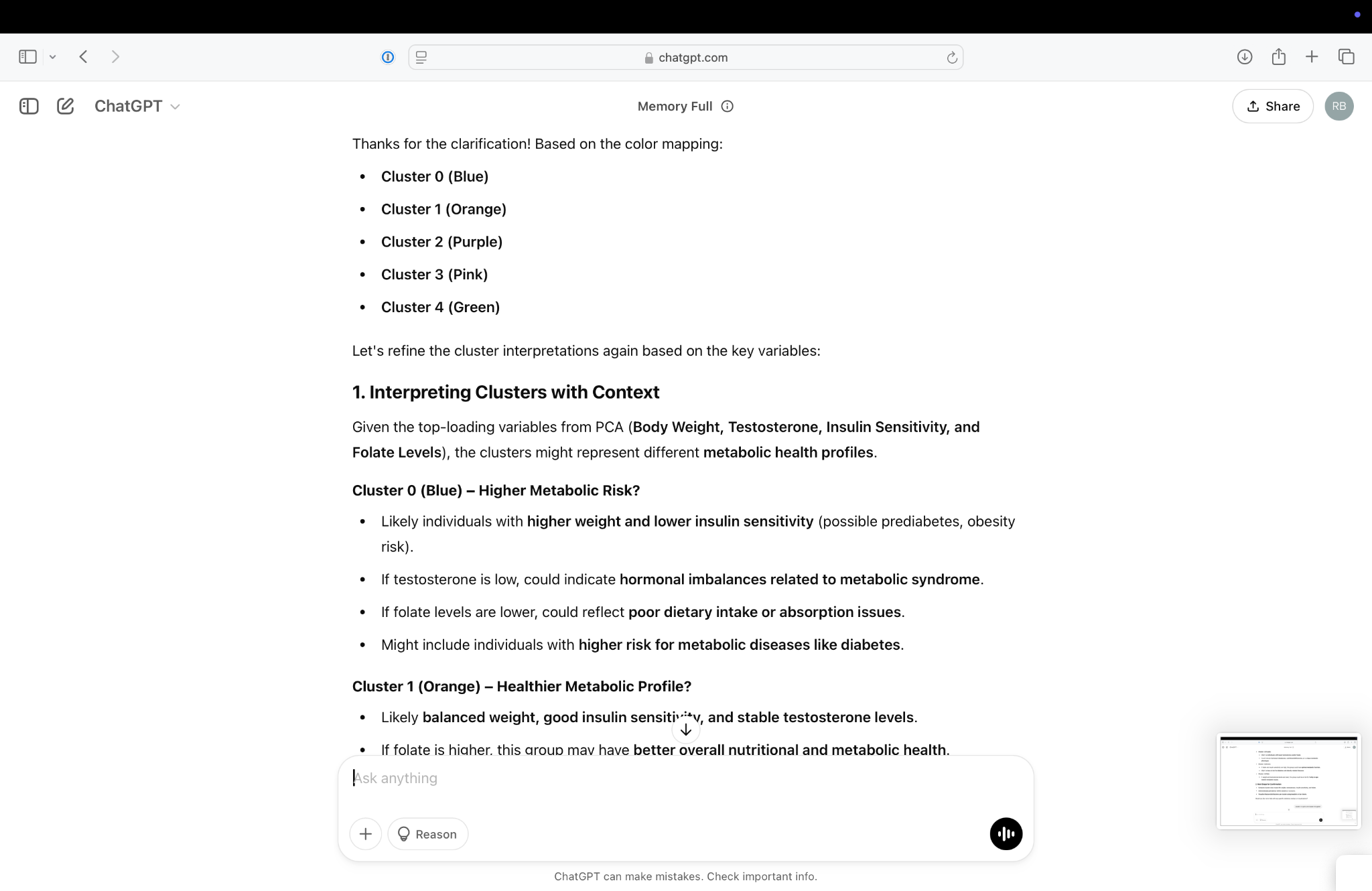
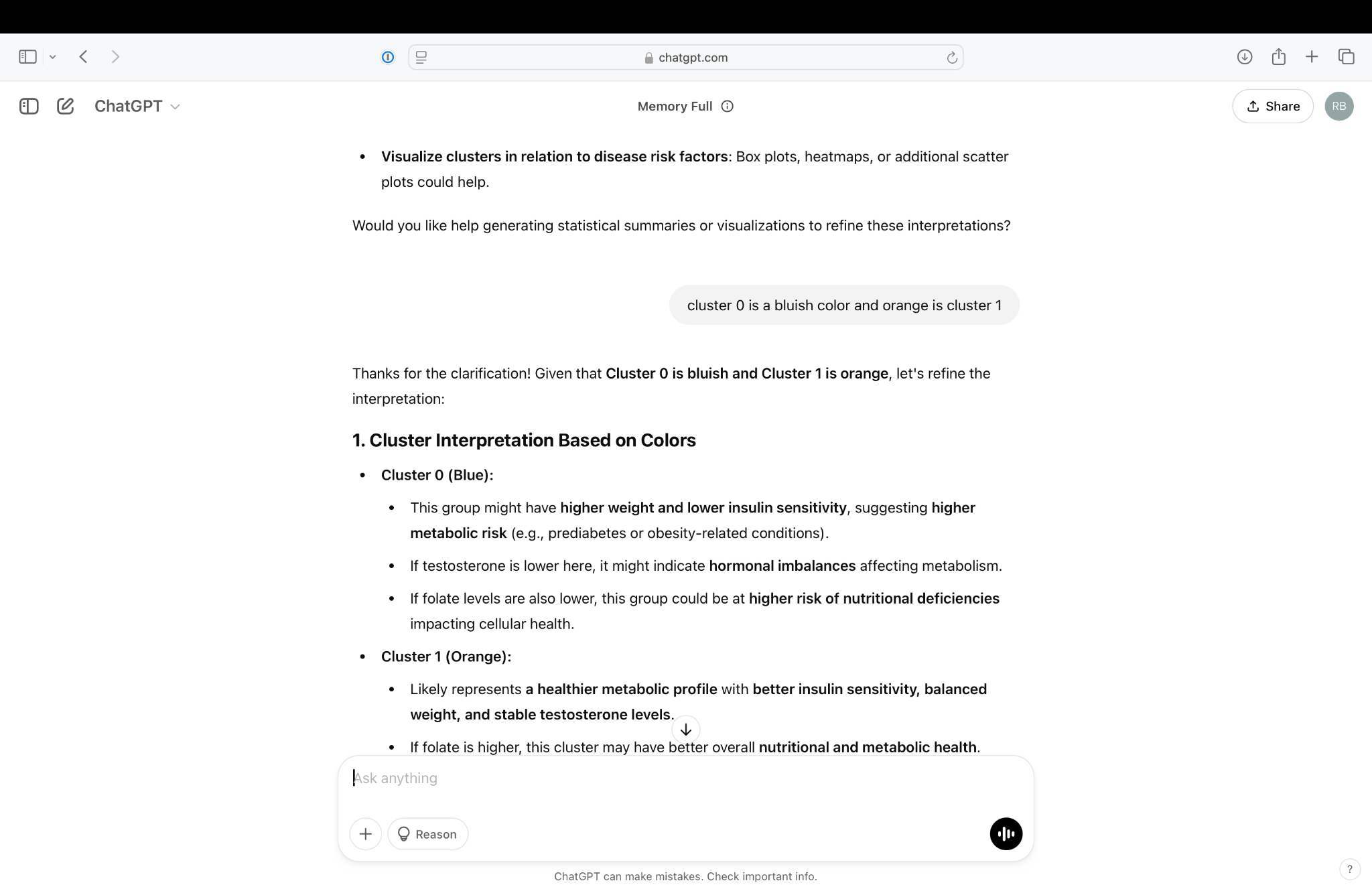
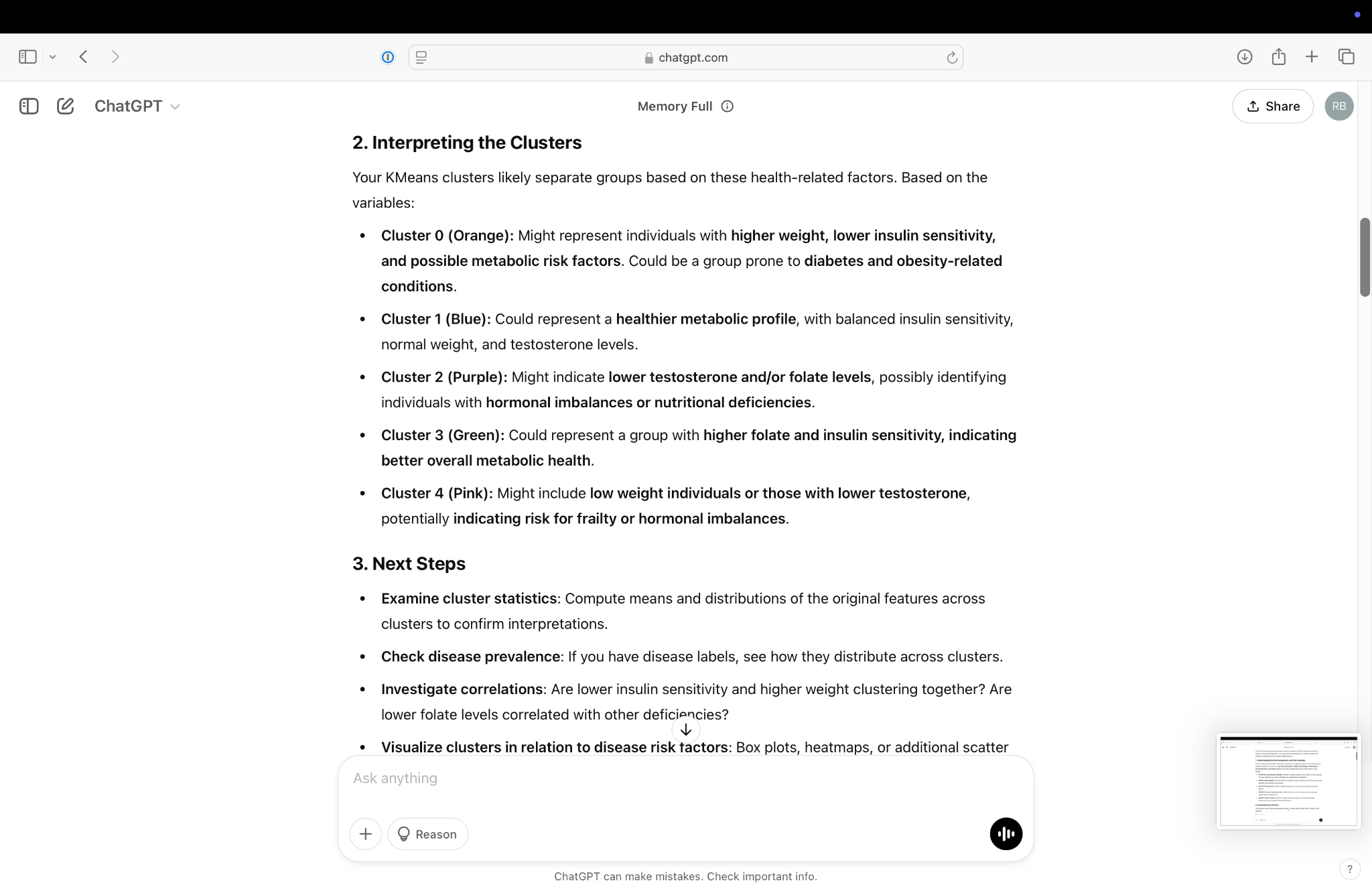
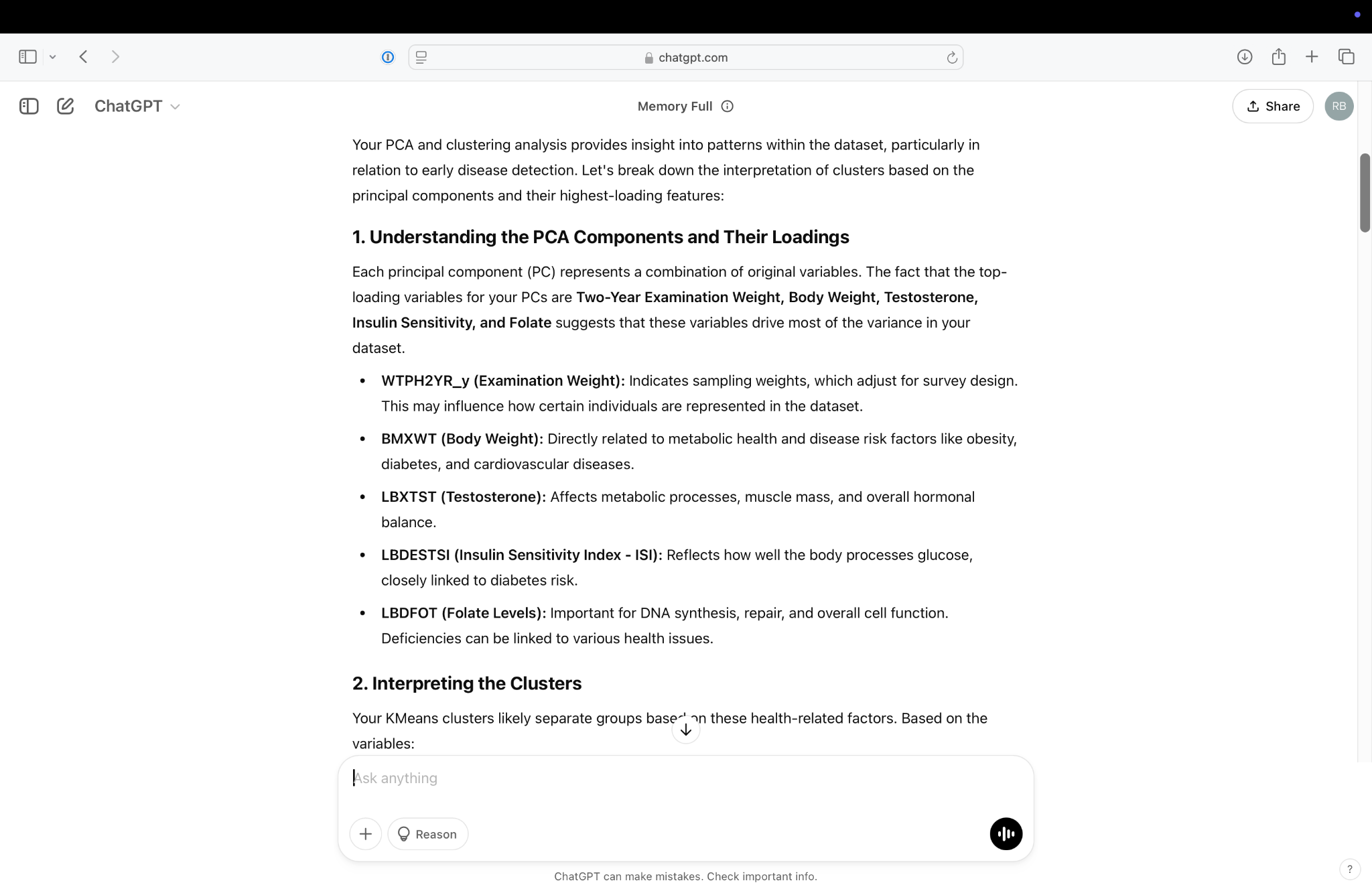
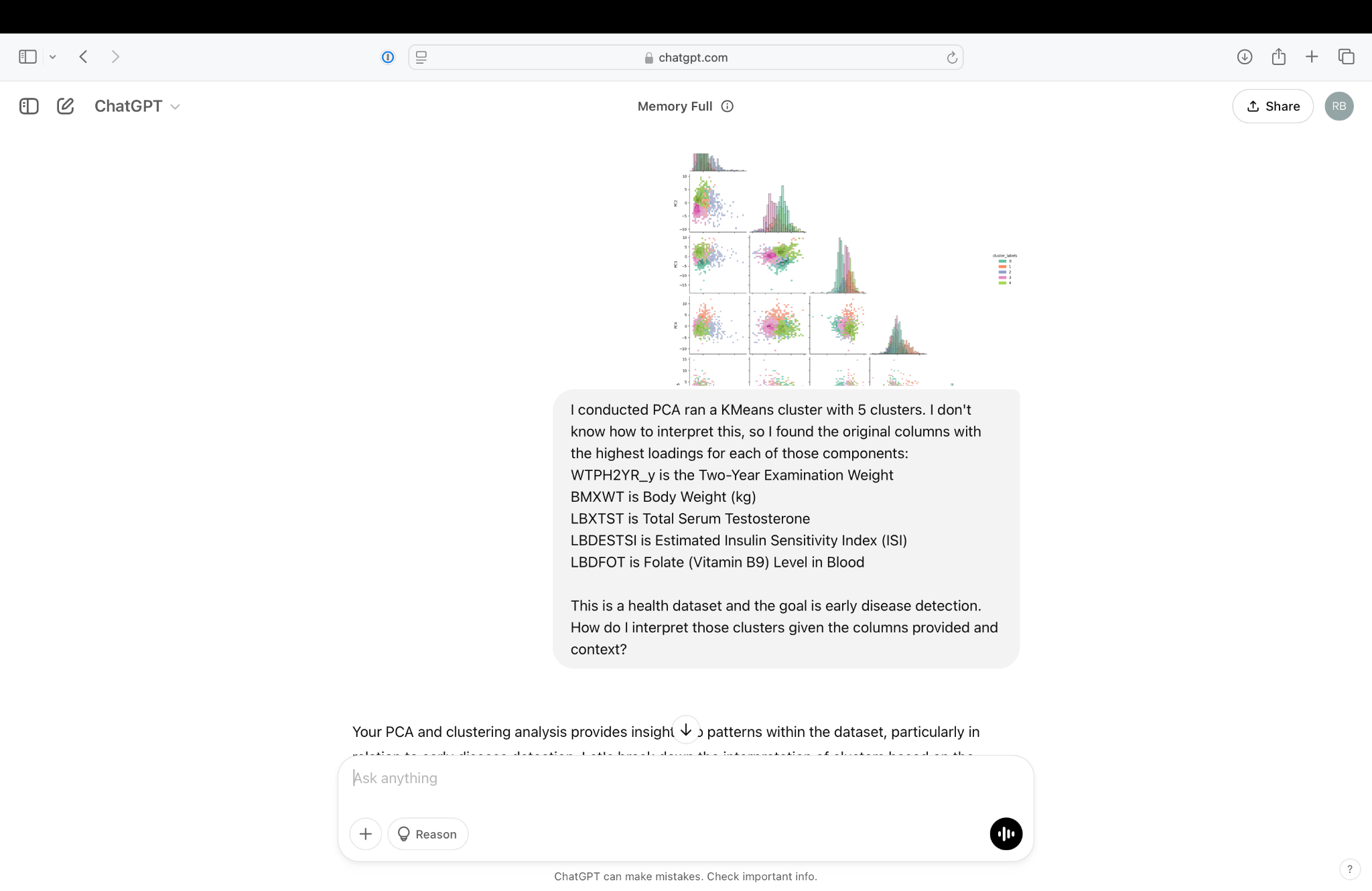
Yashna Meher: Detailed EDA along with network maps and filtering correlations to find group of redundant features, initial Hierarchical clustering using the primary dataset  
Mishil Trivedi: Worked on CUR Approach and followed it with application of it in Hierarchical Clustering, DBSCAN Epsilon, K-Means + t-SNE and the LLM Application on the UMAP

**GitHub Project**:

Repository Link: <https://github.com/sanjal02/Early-Disease-Prediction>

**References**:

1. Mullin, Tara. "DBSCAN Parameter Estimation Using Python." *Medium*, n.d. Accessed March 3, 2025. <https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd>.
2. Dasgupta, Saurabh. "Understanding the Epsilon Parameter of DBSCAN Clustering Algorithm." *Medium*, n.d. Accessed March 3, 2025. <https://medium.com/@saurabh.dasgupta1/understanding-the-epsilon-parameter-of-dbscan-clustering-algorithm-fe85669e0cae>.
3. DataCamp. "A Guide to the DBSCAN Clustering Algorithm." *DataCamp*, October 2024. Accessed March 3, 2025. [https://www.datacamp.com/tutorial/dbscan-clustering-algorithm.](https://www.datacamp.com/tutorial/dbscan-clustering-algorithm)
4. Mahoney, Michael W., and Petros Drineas. "CUR Matrix Decompositions for Improved Data Analysis." *Proceedings of the National Academy of Sciences* 106, no. 3 (2009): 697–702. https://www.pnas.org/doi/10.1073/pnas.0803205106.
5. Shappiron. "CUR Decomposition." *GitHub*, n.d. Accessed March 3, 2025.<https://github.com/shappiron/CUR_decomposition>.
6. Allen, Kenneth J., and S. De Pascuale. "CUR-Based Dynamic Mode Decomposition." *GitHub*, 2021. Accessed March 3, 2025.<https://github.com/KennethJAllen/cur-dynamic-mode-decomposition>.



ChatGPT was used to help with cluster interpretation given the team’s lack of domain knowledge.

**Timeline:**

**Week 1: Data Finding & Ideation**

**Day 1-2:**

* Review previous analysis and define project goals.
* Identify necessary datasets and discuss potential additional data sources.
* Assign roles and responsibilities to each team member.

**Day 3-4:**

* Perform initial exploratory data analysis (EDA) to understand feature distributions.
* Discuss potential dimensionality reduction and clustering techniques.

**Day 5:**

* Finalize the analysis plan based on findings.
* Prepare the dataset for preprocessing.

**Week 2: Data Processing & Dimensionality Reduction**

**Day 6-7:**

* Perform data cleaning (handle missing values, remove duplicates, standardize features).
* Ensure dataset is formatted correctly for further analysis.

**Day 8-9:**

* Apply dimensionality reduction techniques (PCA, alternative methods).
* Compare variance retention and interpretability across methods.

**Day 10:**

* Identify key features retained in the reduced dimensions.
* Document findings and prepare data for clustering.

**Week 3: Clustering Refinement & Validation**

**Day 11-12:**

* Apply clustering methods to the reduced dataset.
* Use the Elbow Method to determine the optimal number of clusters.

**Day 13-14:**

* Validate clustering using silhouette scores and other metrics.
* Analyze cluster stability across multiple runs.

**Day 15:**

* Interpret key features that drive cluster formation.
* Assess clinical relevance of clusters.

**Week 4: Final Report & Presentation Preparation**

**Day 16-17:**

* Compile analysis results into a structured report.
* Interpret clustering results for early disease detection.

**Day 18-19:**

* Develop visualizations and summary insights.
* Prepare final presentation slides.

**Day 20:**

* Conduct internal review, refine presentation, and finalize documentation.

**Supplemental Data and Results:**

The LLM-based method, leveraging Mistral-7B, was implemented as a supplementary approach to analyze medical text data and generate contextual insights. However, being an open-source and free alternative, it is not as well-trained as models like GPT-4o. Due to resource constraints, this was the only feasible method available, and the output had to be limited to prevent session crashes. As a result, while the LLM was included as the last base to be covered, it was not the primary focus of our methodology.

While LLMs can process symptom descriptions and provide interpretations based on prior knowledge, they do not inherently learn from the dataset's numerical structure. Unlike statistical models, they lack direct validation metrics like accuracy or recall unless fine-tuned for specific disease prediction. Without structured labels or integration with numerical features, the LLM remains a text-driven assistant rather than a predictive model for early disease detection. This highlights the need for a specialized medical LLM that is appropriately trained to handle structured numerical data alongside textual insights. Integrating such a model with clustering techniques or supervised learning could enhance predictive performance, enabling a more robust and interpretable framework for disease assessment. However, due to our current computational challenges and highly complex dataset, this method often ended up crashing the session, rendering us unable to draw any analyses from it.