CS 483 - Big Data Mining
Final Project Report - Fall 2024

Instructor: Prof. Lu Cheng

1 Group Information

Group Name: Spaghetti Coders

Group Members: Eleonora Cabai (665321925), Filippo Corna (658181569), Davide Ettori (655693998), Patrik Poggi (651290827)

Group Leader: Filippo Corna

2 Introduction

The intersection of health and Artificial Intelligence has unveiled significant opportunities, in particular in the domain of predictive modeling and decision-making. With this purpose, traditional machine learning algorithms, such as logistic regression and decision trees, have been widely used for medical datasets to predict health outcomes.

In this project, we aim to investigate the predictive power of Machine Learning algorithms on the CDC Diabetes Health Indicators dataset, exploring whether these models can help identify key risk factors and support healthcare decisions.

The aim of our project is to train a neural network to predict the probability of diabetes presence. Then, we analyze the neural network with the SHAP method and extract the interpretability coefficients for each feature. At this point, we pass the patient's data, the neural network's prediction, and the interpretability information to an LLM model. Using this, we generate a medical report for the patient, highlighting the positive and negative aspects of their health and concluding with some recommendations to mitigate risks.

3 Problem and Goal

Diabetes is an increasing public health issue, and early identification of individuals at risk is essential for effective prevention and management. Our primary goal is to classify individuals as healthy, pre-diabetic, or diabetic based on a variety of health, lifestyle, and demographic factors. We also aim to:

- **Identify significant risk factors** contributing to the onset of diabetes, such as physical activity levels and other health indicators.
- **Ensure fairness and unbiased predictions**, particularly across different demographic groups, by assessing the model's performance on sensitive variables like income and education.
- **Conduct cluster analysis** to discover subgroups with distinct health profiles or behaviors, which could reveal underlying patterns in diabetes susceptibility.
- **Develop a predictive health score** that ranks individuals based on their risk of developing diabetes, supporting healthcare professionals in early diagnosis and intervention.

By pursuing these goals, we seek to enhance the understanding of diabetes risk and improve strategies for its prevention and treatment.

4 Dataset Information

In this project, we aim to investigate the predictive power of Machine Learning algorithms on the CDC's Diabetes Health Indicators Dataset, exploring whether these models can help identify key risk factors and support healthcare decisions. Table 1 presents the dataset's features along with their corresponding values or ranges.

Feature	**Values**	**Range**
Diabetes	0 / 1 / 2	-
High Blood Pressure	0 / 1	-
High Cholesterol	0 / 1	-
Cholesterol Check (last 5 years)	0 / 1	-
BMI	-	12 - 98
Smoker	0 / 1	-
Stroke	0 / 1	-
Heart Disease or Attack	0 / 1	-
Physical Activity	0 / 1	-
Fruits	0 / 1	-
Veggies	0 / 1	-
Heavy Alcohol Consumption	0 / 1	-
Healthcare Coverage	0 / 1	-
No Doctor because of Cost	0 / 1	-
General Health	-	1 - 5
Days of not good Mental Health	-	0 - 30
Days of not good Physical Health	-	0 - 30
Difficulties in Walking	0 / 1	-
Sex (female / male)	0 / 1	-
Age category (from 18-24 to 80+)	-	1 - 13
Education level	-	1 - 6
Income level	-	1 - 8

Table 1: Feature Values

5 Dimensionality Reduction

Dimensionality reduction is the process of reducing the number of features in a dataset while preserving important information. This is often achieved through techniques like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), which transform the original high dimensional data into a lower dimensional space. More recently Neural Networks, specifically Autoencoders have also proven to be very a effective method for non-linear Dimensionality Reduction.

In diabetes data analysis, dimensionality reduction can improve computational efficiency, reduce noise, and enhance the interpretability of models by focusing on the most relevant features. This is particularly useful in data mining, where managing large, complex datasets is crucial for effective predictive modeling and pattern discovery.

Each analysis aims at going from the 21 initial features to 3 reduced features, which is the the highest number of dimensions that can be visualized.

5.1 Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimensionality reduction technique that finds directions (principal components) capturing the greatest variance in a dataset. Given a centered data matrix $X \in \mathbb{R}^{n \times d}$ (where each column has zero mean), PCA can be derived using Singular Value Decomposition (SVD). Performing SVD on X gives $X = U\Sigma V^T$, where U and V are matrices of left and right singular vectors, and Σ is a diagonal matrix of singular values. The principal components correspond to the columns of V, ordered by the magnitude of singular values in Σ. To reduce dimensionality, we project X onto the first k components using XV_k, where V_k includes the first k columns of V. This projection captures the maximum variance, minimizing reconstruction error, but it has only access to linear transformations of the features. An important advantage of this method is that we do not have to choose any hyperparameters, beside the number of components. Finally, as we will see, it's naturally interpretable, since when can clearly see the contribution of each feature to each Principal Component.

The PCA analysis makes the dataset easily visualizable in three dimensions. We can observe that subjects without diabetes are clustered to the left (purple), while those with diabetes are more towards the right (yellow), as shown in Figure 1. The three most influential Principal Components capture, respectively, 16.3%, 8.5%, and 6.3% of the total data variance, revealing over 31% of the variance with only three features (instead of 21). This reduction can be utilized later to train more precise models that are less affected by noise present in the data.

Finally, we note that the reconstruction error, measured by MAE (Mean Absolute Error), is 0.59. This represents the average absolute error per feature when attempting to return to the 21-feature representation from the three selected components.

Regarding interpretability, we can analyze the contributions of each feature to each component, as shown in the plots below. For instance, we observe that the first component represents medical features: General Health, Physical Health, and Difficulty in Walking. The second component reflects the patient's social situation: Age, ability to see a doctor, and healthcare coverage. The third component focuses on lifestyle habits, such as frequency of eating fruits, vegetables, and smoking. These insights align with our expectations and already offer an interesting view of the situation, as well as providing a method for dimensionality reduction for future analyses.

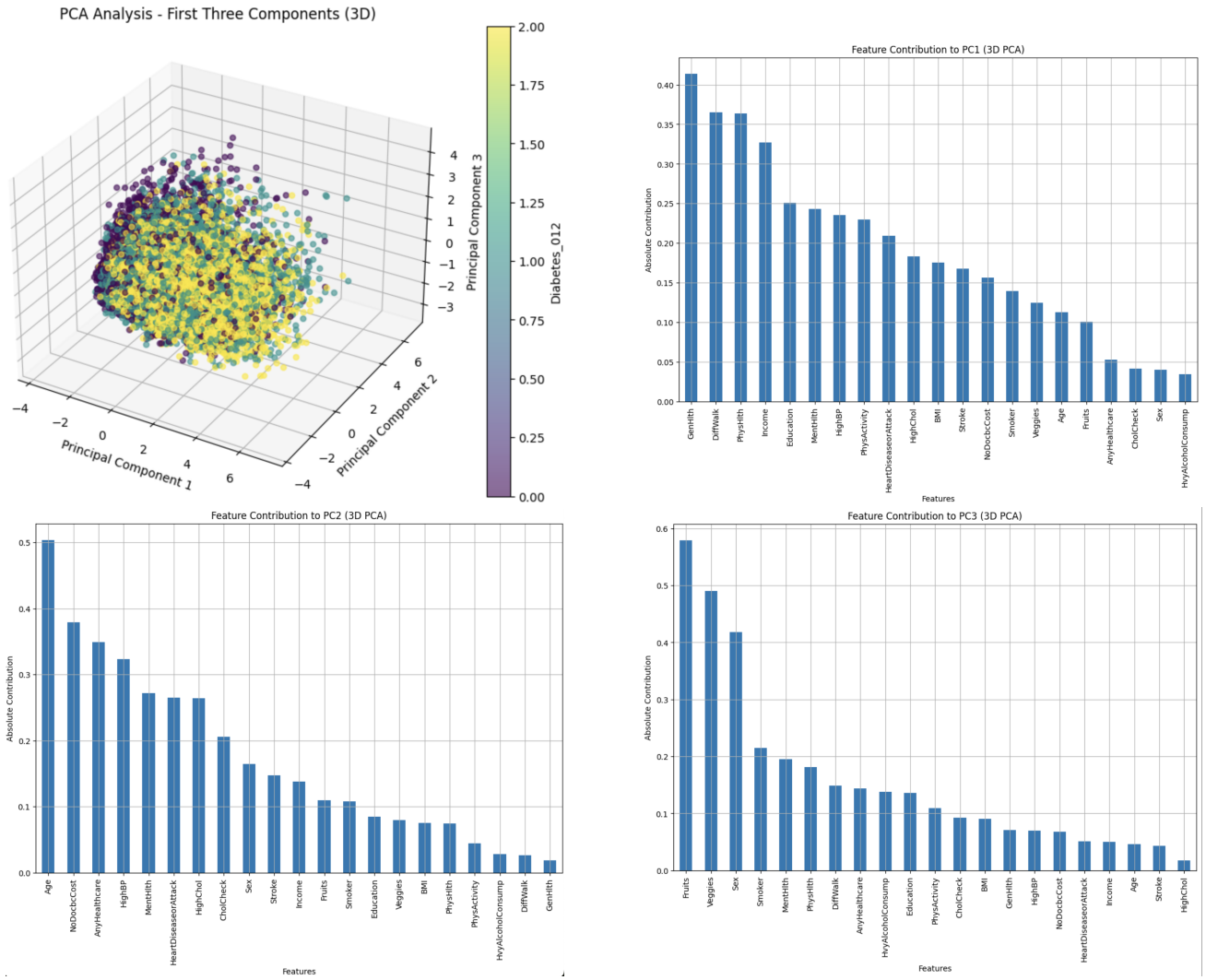


Figure 1: PCA Data Visualization and individual Feature Contributions

5.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a non-linear dimensionality reduction technique designed to visualize high-dimensional data by preserving local similarities. The main idea is to model the probability of similarity between points in high-dimensional space and map it to a lower-dimensional space while maintaining these relationships. For points x_i and x_j in the original space, t-SNE calculates the probability p_{ij} of x_j being a neighbor of x_i using a Gaussian distribution centered on x_i. In the low-dimensional space, the probability q_{ij} is calculated similarly, but with a Student's t-distribution to handle crowding. The objective is to minimize the Kullback-Leibler (KL) divergence between p_{ij} and q_{ij}, defined as:

$$\mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

This ensures that nearby points in the high-dimensional space remain close in the lower-dimensional embedding, creating a visual representation that captures local structure. The **perplexity** is the most important hyperparameter and it balances the focus between local and global aspects of the data. Lower perplexity values emphasize local relationships, while higher values capture broader, more global structures.

The t-SNE analysis shows the data clustered in a less regular manner, which results from the non-linearity of this probability-based method. One of the main challenges lies in finding the right value for the perplexity

parameter, which depends on the specific context and is not easy to determine a priori. In this case, we conducted an exhaustive search across all values from 1 to 100 in intervals of 10. The best value turned out to be 70, excluding the value 1, which seems to represent an overfitting case, overly focused on local characteristics of the data. Unfortunately this method does not permit us to analyze the captured variance or the reconstruction error, since once compressed the data cannot be brought back to the original dimensions.

Finally, we observe that no particular pattern seems to emerge in the data, suggesting that t-SNE might not be well-suited for this specific problem or data distribution, even if it's in theory more general than PCA. This is also evidenced by the final Kullback-Leibler Divergence value of 1.28, whereas a well-executed t-SNE analysis usually achieves a value below 1.

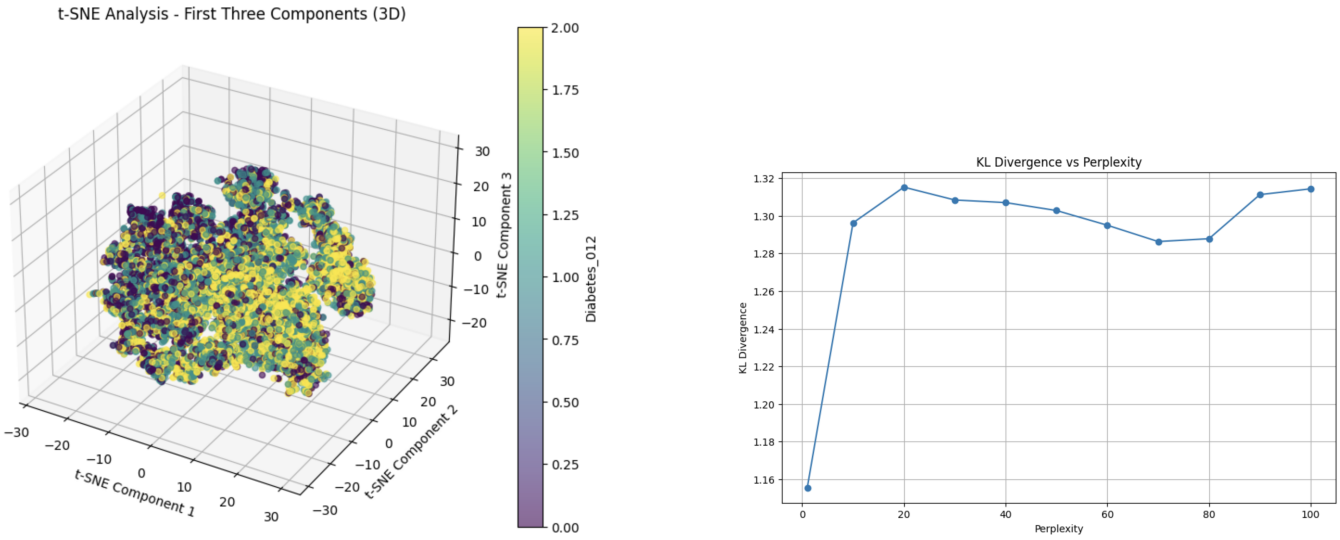


Figure 2: t-SNE Visualization (Left) and KL Divergence Variation with Perplexity (Right)

5.3 Autoencoders

Autoencoders are neural networks used for dimensionality reduction by learning efficient representations of data. They consist of an encoder, which maps input x to a lower-dimensional latent space z, and a decoder, which reconstructs x from z, aiming to minimize reconstruction error. Given an input x, the encoder function $f(x) = z$ and decoder function $g(z) = \hat{x}$ are optimized to minimize $L(x, \hat{x})$, typically using mean squared error. By constraining the latent dimension z to be smaller than the input, autoencoders capture essential data features. This enables denoising, compression, and effective dimensionality reduction, especially in cases where linear methods like PCA are insufficient. It's a very powerful method and being a Neural Network it inherits all their typical strength (very general) and weaknesses (high variance).

The autoencoder consists of a neural network with the following layers: 21, 16, 8, 3, 8, 16, 21. The activation functions are Leaky ReLU, the loss function is mean squared error, and the optimizer is Adam. We observe that the training converges after approximately 30 epochs, and the process is then stopped at the 40th epoch.

We note that the data distribution is entirely different from the previous methods, with multiple clusters scattered in the 3D space. This results from the neural network's ability to model highly complex shapes, thanks to its structure combining linear layers and non-linear functions. Here, we observe a much clearer separation compared to previous methods and a significantly lower reconstruction error, with a Mean Absolute Error (MAE) of 0.47. This indicates that the reduction has been quite successful and that this representation effectively captures the true characteristics of the data. The downside is that using a neural network has resulted in a loss of interpretability, as these architectures are nearly black-box models.

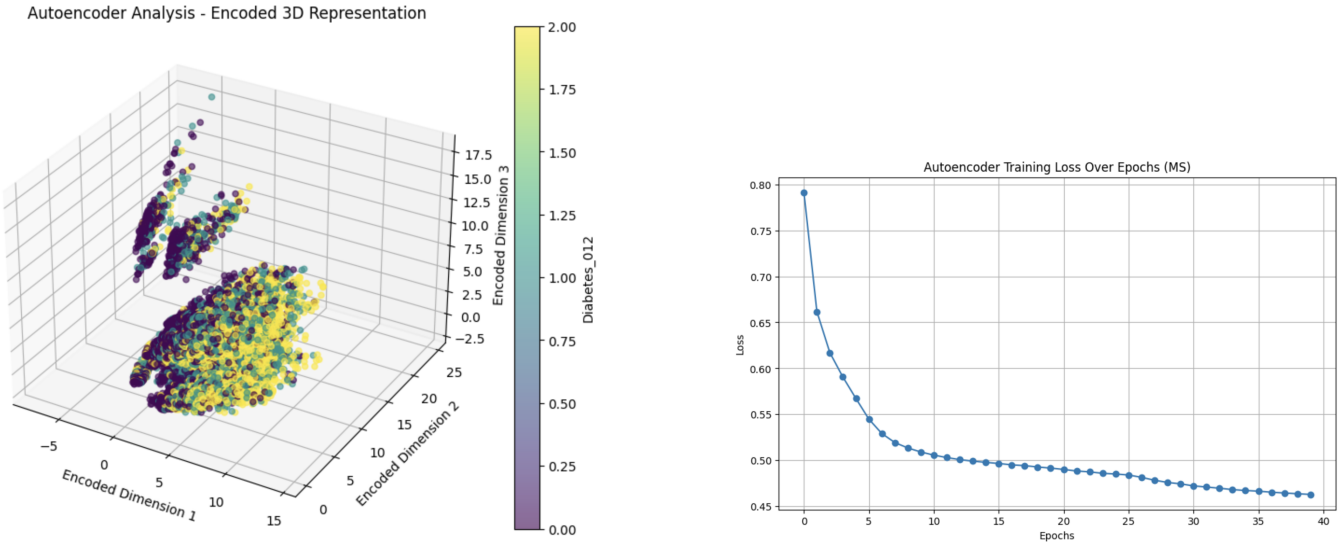


Figure 3: Left: Compressed Data with Autoencoder; Right: Loss per Epoch

6 K-means Clustering Analysis

Clustering is an unsupervised learning technique used to group data points into clusters that share similar characteristics. K-means clustering partitions data by minimizing the within-cluster variance, a process governed by the number of clusters k. Each data point is assigned to the nearest cluster center, iteratively refining these centers until convergence. The goal is to assign similar health profiles to the same cluster.

6.1 K-Means Clustering for Health Profile Identification

In this study, we utilized K-means clustering to identify distinct health profiles within the dataset, focusing on uncovering underlying patterns that may contribute to diabetes susceptibility. Specifically, we applied two versions of K-means clustering: traditional K-means and Stratified K-means.

6.1.1 Traditional K-Means Clustering

Traditional K-means clustering was employed to divide the dataset into three distinct groups based solely on feature similarity, without any prior knowledge of diabetes status. This clustering approach aimed to reveal natural groupings within the dataset, potentially illuminating health profiles that might be more susceptible to, or protected from, diabetes.

The resulting clusters show clear differences in average feature values across several health indicators, as outlined in Table 2. For instance, individuals in Cluster 1 exhibited higher BMI and a greater likelihood of having high blood pressure, high cholesterol, and a history of stroke, which are well-established diabetes risk factors. Meanwhile, Clusters 0 and 2 displayed lower average BMI, fewer incidences of high blood pressure, and generally more active lifestyles. These groupings provide insight into distinct health profiles within the population and underscore certain features associated with potential risk factors for diabetes.

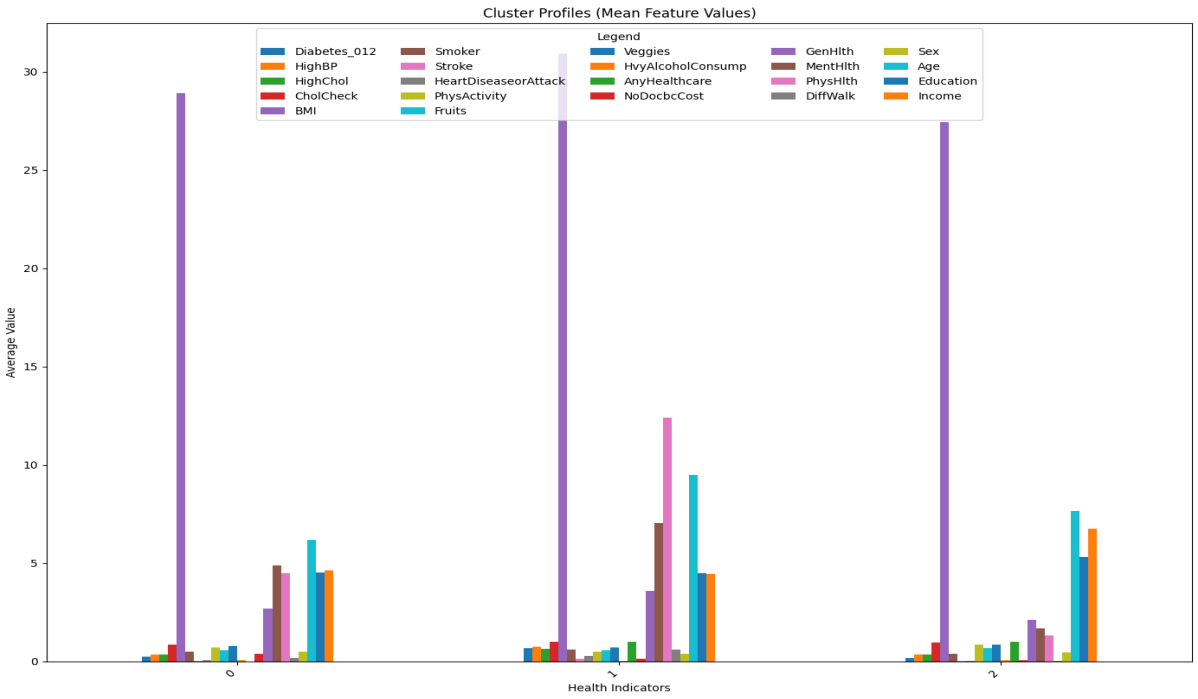


Figure 4: Cluster Profiles (Mean Feature Values)

Feature	Cluster 0	Cluster 1	Cluster 2
Diabetes	0.2456	0.6531	0.1727
High BP	0.3428	0.7332	0.3259
High Chol	0.3298	0.6387	0.3537
Chol Check	0.8639	0.9847	0.9617
BMI	28.8975	30.9095	27.4400
Smoker	0.4938	0.6070	0.3809
Stroke	0.0302	0.1290	0.0096
Heart Disease	0.0673	0.2730	0.0319
Phys Activity	0.6912	0.4950	0.8549
Fruits	0.5677	0.5444	0.6711
Veggies	0.7612	0.6980	0.8556
Hvy Alcohol	0.0670	0.0354	0.0629
Healthcare	0.0000	0.9992	1.0000
No Doc bc Cost	0.3698	0.1441	0.0428
Gen Hlth	2.6959	3.5942	2.1102
Ment Hlth	4.8715	7.0258	1.6896
Phys Hlth	4.4705	12.3878	1.3044
Diff Walk	0.1533	0.5796	0.0217
Sex	0.4835	0.3930	0.4543
Age	6.1581	9.4977	7.6370
Education	4.5204	4.4747	5.2938
Income	4.6165	4.4295	6.7366

Table 2: Cluster Profiles (Mean Feature Values)

6.1.2 Stratified K-Means Clustering

The second approach, Stratified K-means, involved clustering based explicitly on diabetes status: Cluster 0 for no diabetes, Cluster 1 for pre-diabetes, and Cluster 2 for diabetes. This stratified approach offers a focused comparison of feature averages across diabetes categories, allowing for an in-depth analysis of how each health indicator varies between non-diabetic, pre-diabetic, and diabetic populations.

Table 3 provides a summary of the average feature values across the diabetes status clusters. Results indicate that diabetes status correlates strongly with several key features: BMI, physical activity, general health , and age. Cluster 2 (diabetes group) showed significantly higher average BMI, lower physical activity, and worse self-reported general health compared to the other clusters. Additionally, it had a higher proportion of individuals with high cholesterol, high blood pressure, and lower income and education levels, further reinforcing established risk factors in diabetes research.

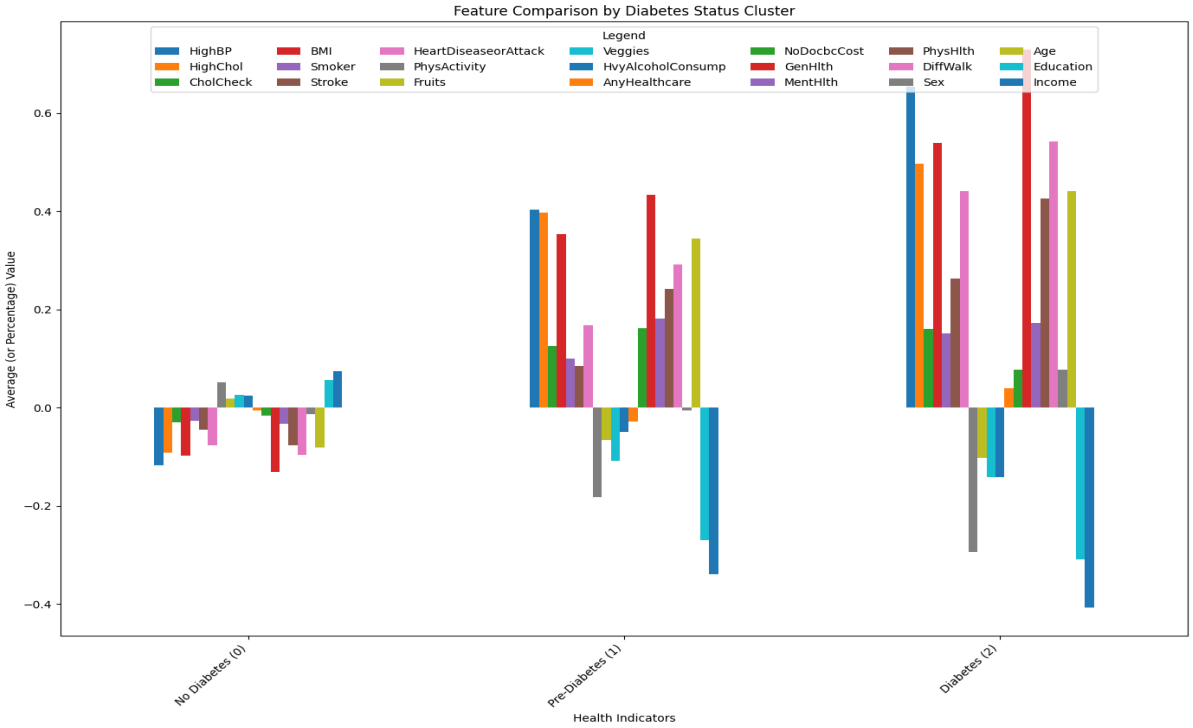


Figure 5: Features Comparison by Diabetes Status

Feature	No Diabetes	Pre-Diabetes	Diabetes
High BP	-0.1169	0.4041	0.6540
High Chol	-0.0910	0.3980	0.4978
Chol Check	-0.0294	0.1263	0.1610
BMI	-0.0968	0.3544	0.5389
Smoker	-0.0272	0.0998	0.1511
Stroke	-0.0453	0.0844	0.2630
Heart Disease	-0.0765	0.1684	0.4406
Phys Activity	0.0525	-0.1819	-0.2936
Fruits	0.0182	-0.0665	-0.1014
Veggies	0.0256	-0.1086	-0.1406
Hvy Alcohol	0.0245	-0.0490	-0.1418
Healthcare	-0.0061	-0.0273	0.0404
No Doc bc Cost	-0.0164	0.1627	0.0781
Gen Hlth	-0.1301	0.4345	0.7296
Ment Hlth	-0.0324	0.1815	0.1723
Phys Hlth	-0.0757	0.2416	0.4258
Diff Walk	-0.0961	0.2921	0.5427
Sex	-0.0128	-0.0053	0.0781
Age	-0.0804	0.3442	0.4410
Education	0.0570	-0.2698	-0.3093
Income	0.0747	-0.3393	-0.4074

Table 3: Feature Comparison by Diabetes Status

6.2 Cluster Profiles and Feature Analysis

Across both clustering methods, we observed several important patterns. In the traditional K-means clusters, diabetes status was not a distinguishing factor, allowing health characteristics alone to dictate group membership. In contrast, the stratified clusters clearly highlighted the differences between individuals at various stages of diabetes. These insights not only underscore the importance of health profiles in diabetes prevention but also support the development of targeted interventions based on health indicators that show strong associations with diabetes progression.

6.3 Feature Distributions by Cluster

To further interpret these clusters, we analyzed the distribution of each feature within them. The distributions, visualized as side-by-side histograms, provide insights into the unique characteristics of each group. For instance:

- Blood Pressure and Cholesterol: Clusters associated with pre-diabetes and diabetes show higher frequencies of elevated blood pressure and cholesterol levels.
- Lifestyle Factors: Features such as physical activity, fruit and vegetable consumption, and smoking habits differ noticeably between clusters, indicating a correlation between these lifestyle choices and diabetes risk.
- Demographics: Age and education levels also vary across clusters, with Cluster 2 (diabetes) showing a higher proportion of older individuals.

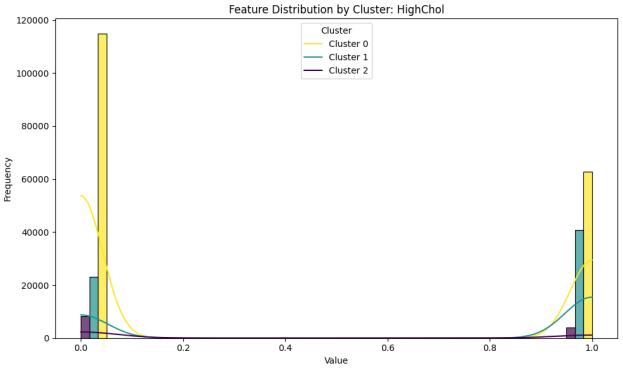


Figure 6: High Cholesterol Activity Histogram

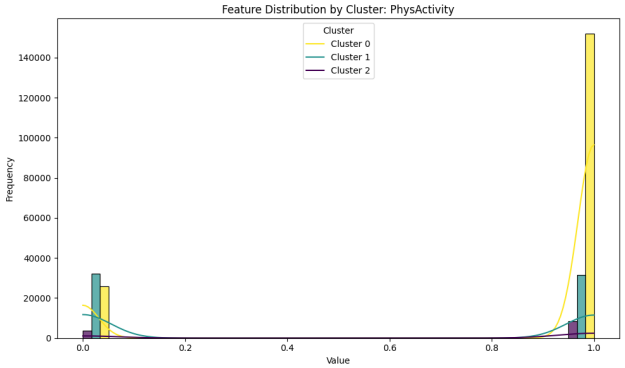


Figure 7: Physical Activity Histogram

6.4 Conclusion

In summary, the K-means clustering analysis provides valuable insights into the dataset, highlighting the diverse health profiles associated with different diabetes susceptibility levels. This information could be useful for identifying high-risk groups and informing targeted prevention efforts based on lifestyle and health indicators.

7 Tree Based Classification

7.1 Introduction

In this section we will discuss a classificatoin technique different from K-Means, based on decision trees. We will explore different methods to apply this theoretical model and investigate different parametrizations for each of them.

7.2 Decision Trees

Decision Trees are the core of Tree Based Classification as they represent the main theoretical model behind this class of machine learning methods.

Decision trees are buit by choosing features splits in order to maxmize information gains as the decision proceeds from the root to the leaves. Different parametrizations can hence lead to different performances.

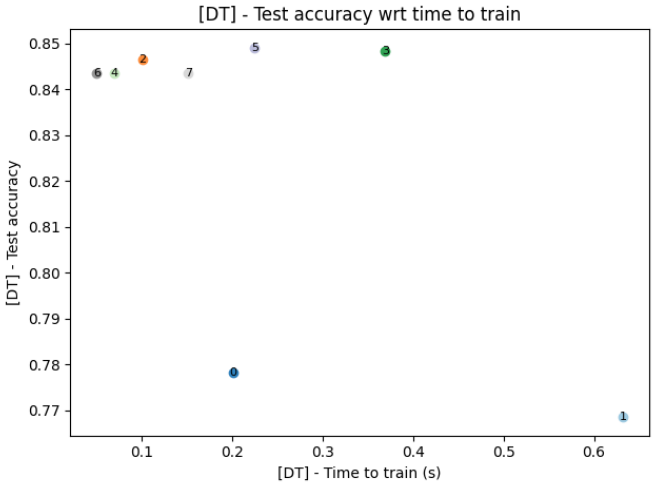


Figure 8: Test Accuracy vs Time to Train

In figure 8 we can observe the results obtained when changing parameters in the Decision Tree classifier under an **accuracy** point of view. This setting is meant to highlight a trade-off: that of accuracy (efficacy) versus time required to train the model (efficiency).

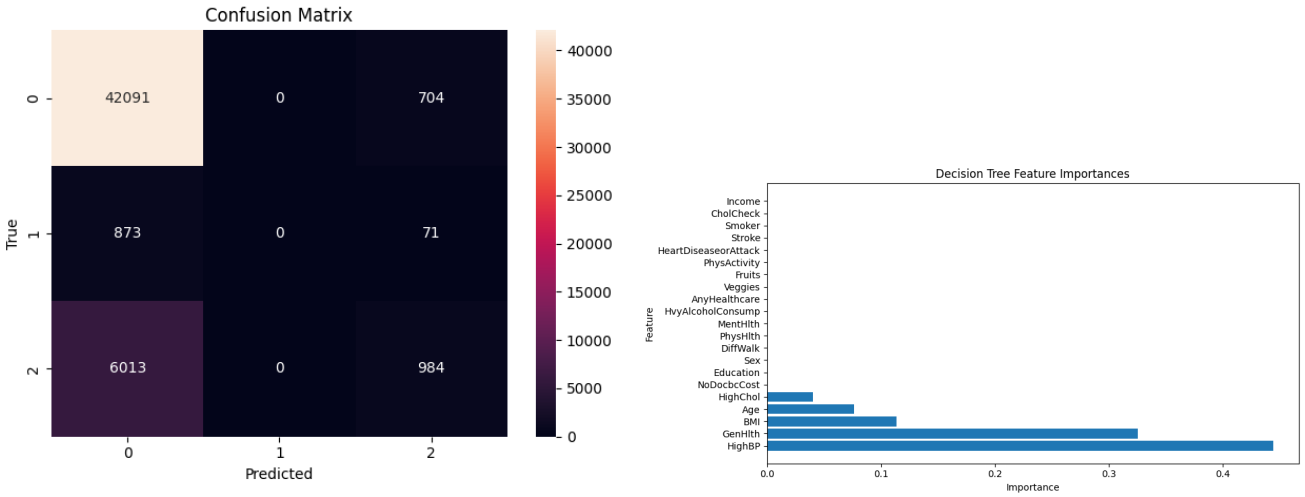
The best accuracy is achieved by setting number 5, with parameters shown in table 4

Setting Code	Max Depth	Max Features	Random State
5	5	21	42

Table 4: DT: setting achieving optimal accuracy

Where, in particular, the number of features is not constrained and we are allowing the model to use all 21 of them. the random state has been set to 42 as in the K-Means discussion for reporting consistency purposes.

The most accurate setting, setting number 5, achieves 84.9% accuracy and presents the confusion matrix depicted in figure 9a



(a) Setting 5 Confusion Matrix **(b)** feature importances in best performing Decision Tree

Figure 9: Best performing Decision Tree and its feature importances

However, since we are dealing with an healthcare-oriented problem, we might be more interested in **recall** rather than in accuracy. This is due to the fact that recall is defined as $\frac{TruePositives}{TruePositives+FalseNegatives}$. Hence, maximizing recall helps achieving the highest possible number of ill people being identified.

As a conseuqence of this, we have designed a linear combination of *recall, f1-score, precision* and *accuracy* in order to obtain a single metric that gives the desired importance to each one of them, prioritizing recall. The formula is the following:

$$\text{score} = 0.4 \cdot \text{recall} + 0.3 \cdot \text{f1-score} + 0.2 \cdot \text{precision} + 0.1 \cdot \text{accuracy}$$

As it turns out, setting 5 (see its confusion matrix in figure 9a) is the best one also in this case.

However, already from the the confusion matrix we can notice two important things:

1. The model is never predicting class 1, which corresponds to 'prediabetic'. This is not a good thing in terms of diagnosis, as we are not able to identify the people that are at risk of developing diabetes.
2. Our model is predicting class 0 very frequently. Most importantly it is misclassifying 6013 instances from class 2 ('diabetic') as class 0. This is a very high number of false negatives, which is not good in terms of diagnosis.

In conclusion even the best performing Decision Tree is not performing sufficiently well given the importance of the problem at hand.

7.3 Bagging and Boosting Tree-Based classification: Random Forest and ADABoost

After decision trees, we have investigated two ensemble methods based on trees: Random Forest and AdaBoost. After a parameter space exploration carried out following the same approach as for Decision Trees, we have concluded that they yield very similar performances.

Therefore, we prefer Decision Trees as they are lighter and significantly easier to interpretet, and we believe that interpretability is a crucial aspect in the healthcare domain. Figure 9b shows the feature importances of the best performing Decision Tree (setting 5, as in table 4).

7.4 Conclusion

Finally we can conclude that while ensemble methods are typically better even in conditions of equal accuracy as they mitigate bias/variance issues, here Decision Trees have been preferred due to their explainability and low overfitting risk (the best performing Decision Tree has a depth of 5).

8 Neural Network

In our initial attempts, we designed neural networks trained on imbalanced datasets. While these models sometimes achieved high accuracy, their recall for underrepresented classes was poor, meaning they often failed to correctly identify less common cases, such as diabetic patients. To address this, we chose a smaller but balanced dataset, evenly split 50% healthy and 50% diabetic individuals, to ensure consistent performance across both classes.

8.1 Neural Network Configuration

8.1.1 Network Architecture

The model architecture consists of a fully connected network with the following structure:

- **Input Layer:** 64 neurons with ReLU activation. ReLU introduces non-linearity, helping the model capture complex patterns in the data.
- **Hidden Layer 1:** 32 neurons with ReLU activation, followed by Batch Normalization to improve training stability and speed.
- **Hidden Layer 2:** 16 neurons with ReLU activation, followed by 30% Dropout and Batch Normalization to prevent overfitting and ensure robust learning.
- **Output Layer:** 1 neuron with Sigmoid activation, providing probability scores between 0 and 1, which allow for flexible interpretation of a patient's risk level.

We opted for this simple architecture because deeper networks with additional layers led to overfitting, resulting in worse performance on the validation set. The dataset's limited complexity makes this streamlined approach more effective.

8.1.2 Model Compilation

The model was compiled using:

- **Optimizer:** Adam, a dynamic optimization algorithm that adjusts learning rates for efficient parameter updates.
- **Loss Function:** Binary Crossentropy, specifically designed for binary classification tasks.
- **Metrics:** Accuracy, to evaluate the model's performance.

8.2 Model Training

8.2.1 Hyperparameters

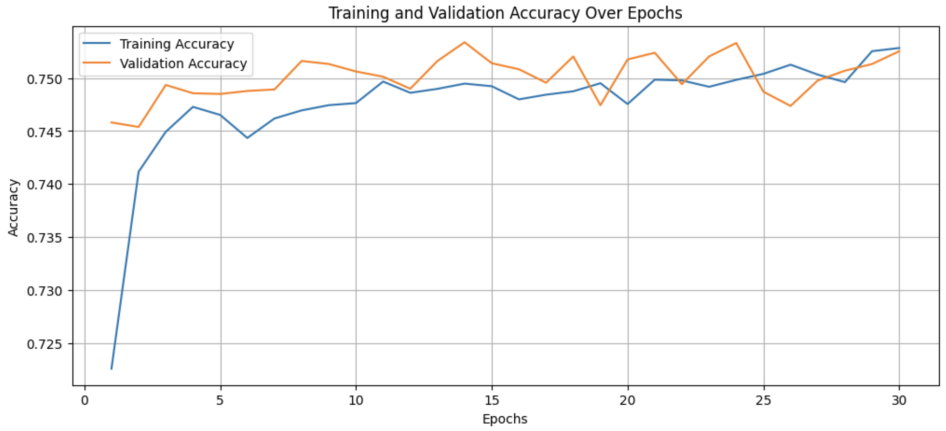
- **Epochs:** 30.
- **Batch Size:** 32.
- **Validation Split:** 20% of the training data was reserved for validation.

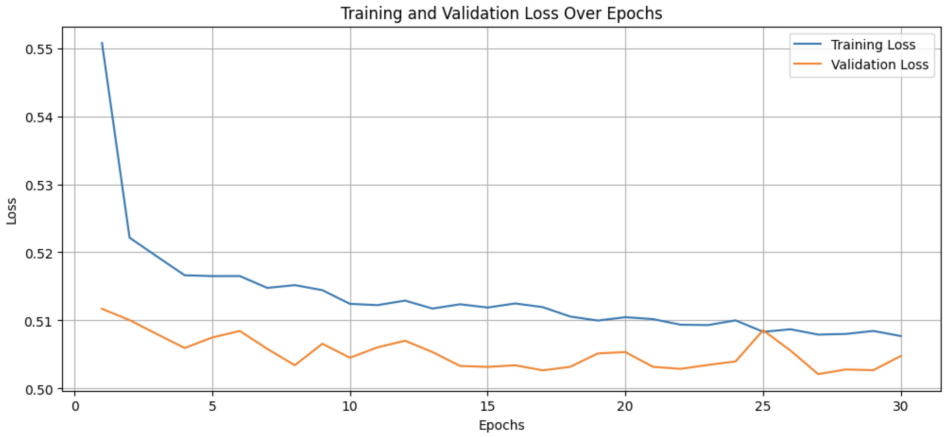
8.2.2 Training Progress

The following observations were made during training:

- **Initial Phase (1–10 epochs):** Training accuracy increased rapidly, showing the model's ability to learn general patterns. Validation accuracy stabilized early, indicating good generalization.

- **Stabilization Phase (10–30 epochs):** Both accuracy and loss curves plateaued, reflecting a balance between learning and generalization without overfitting.





8.3 Model Evaluation

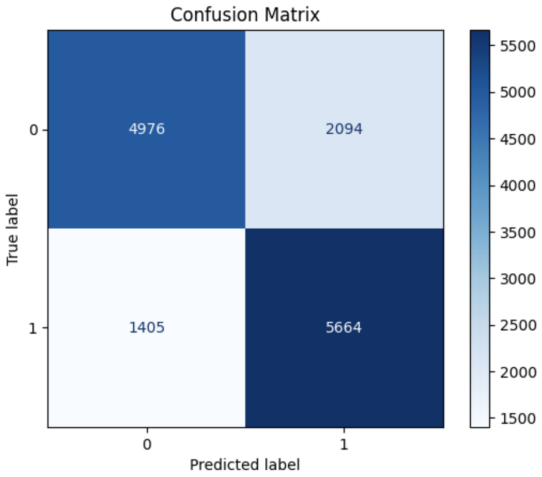
8.3.1 Final Accuracy

The model achieved a test accuracy of 75.25% with a final loss of 0.5048, indicating good overall classification performance.

8.3.2 Confusion Matrix

The confusion matrix highlights the model's performance:

- **Class 0 (healthy individuals):** 4976 samples correctly classified, 2094 misclassified.
- **Class 1 (diabetic individuals):** 5664 samples correctly classified, 1405 misclassified.



8.3.3 Classification Report

Class	Precision	Recall	F1-Score
Class 0	78%	70%	74%
Class 1	73%	80%	76%

Table 5: Classification report for the neural network model.

The weighted average F1-Score is 75%, showing balanced performance across both classes.

8.4 Conclusions

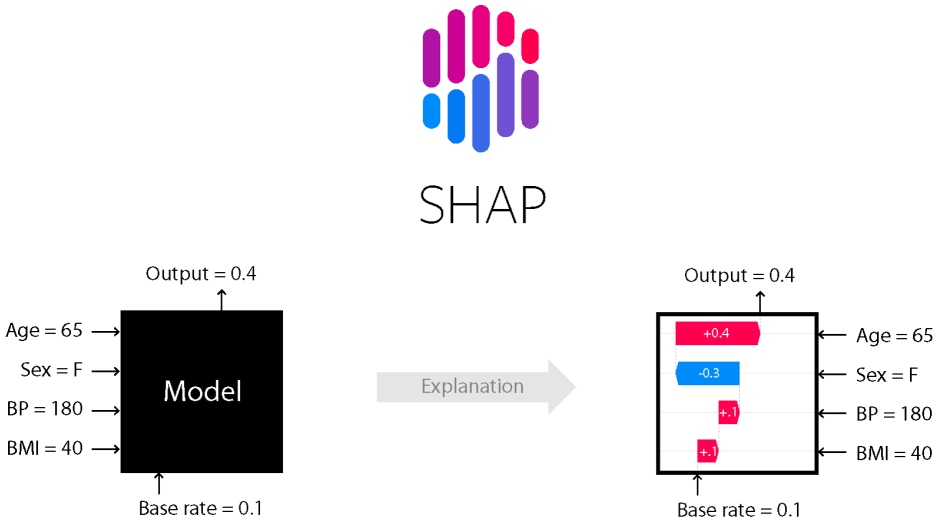
The model demonstrates strong overall performance, successfully balancing precision and recall across both classes. However, there is room for improvement, particularly in refining predictions for diabetic individuals (Class 1). Based on our observations, here are potential areas for enhancement:

- **Feature Engineering:** Introduce new features or transform existing ones to provide the model with richer input data, potentially improving classification performance.
- **Hyperparameter Tuning:** Explore different learning rates, batch sizes, or dropout rates to fine-tune the model further.
- **Data Augmentation:** Although the dataset is balanced, augmenting the data (e.g., generating synthetic samples) could help capture more complex patterns, particularly for edge cases.

This simplified architecture, combined with the balanced dataset, represents a significant improvement over previous attempts, ensuring reliable and consistent predictions for both healthy and diabetic individuals.

9 Interpretability (SHAP)

Neural networks (NNs) are powerful tools for modeling complex relationships in data. However, their black-box nature makes it challenging to understand the rationale behind their predictions, especially in high-stakes fields such as healthcare. SHAP (SHapley Additive exPlanations) is a method rooted in game theory that provides a unified framework for interpreting machine learning models, including neural networks. SHAP is based on the concept of Shapley values, which originate from cooperative game theory.



The Shapley value is a way of fairly distributing the total gain (or cost) of a game among its participants (or players). In the context of machine learning, the "players" are the features of the input data, and the "gain" corresponds to the model's prediction. The Shapley value of a feature represents its contribution to the prediction, calculated by considering all possible subsets of features and marginalizing over their contributions. The sign of a Shapley value indicates whether the feature increases or decreases the predicted outcome. A positive value means the feature contributes positively, pushing the prediction higher, while a negative value indicates a negative contribution. The magnitude of the Shapley value quantifies the strength of this contribution, with larger absolute values signifying greater influence. This property makes SHAP a powerful tool for explaining individual predictions by attributing importance to each feature in a manner consistent with the underlying model. Applying SHAP to neural networks enables the extraction of feature importance rankings, which can help refine models and improve their interpretability. However, while SHAP is highly insightful, it has some limitations. Calculating exact Shapley values is computationally expensive, particularly for complex models like deep neural networks, as it requires evaluating all possible feature combinations. Approximations, such as those provided by many SHAP implementations, help mitigate this issue but may introduce inaccuracies. Additionally, interpreting Shapley values still requires domain knowledge to ensure that the explanations align with real expectations. In our case, we use SHAP as follows. First, we fit SHAP on our dataset in question, using 250 data points as the baseline and ensuring that SHAP can adjust itself to the realistic distribution of the data it will receive. At this point, for each new patient, their data is passed first through the neural network and then through SHAP, in order to obtain both the prediction and a list of contributions (numerical coefficients) for each of the patient's features. Of course, before passing the data to SHAP (both the baseline data and the new ones), the same preprocessing done by the neural network (Min Max Scaling) must be applied to ensure that the scale and distribution of the values are consistent. Once the SHAP values for each feature are obtained, they are sorted, and we select the 3 most negative (Bad Features) and the 3 most positive (Good Features), along with the mean of the absolute values. This information is then provided to the LLM, which uses it, along with the complete features and the neural network's prediction, to generate a medical report about the patient. Note that, in addition to the names of the 3 Bad Features and 3 Good Features, their SHAP values in absolute terms are also provided.

10 LLM Integration

Large Language Models (LLMs) represent a significant advancement in natural language processing. Their ability to generate coherent and contextually relevant text makes them invaluable for tasks requiring complex communication and explanation. In this project, an LLM was integrated to improve diabetes predictions by generating professional and patient-specific reports. The generated report serves multiple purposes:

- **Patient Communication:** the LLM translates the technical results into easily understandable language, enabling patients to understand their health status;
- **Medical Insights:** the report supports healthcare providers in validating predictions and tailoring interventions;
- **Actionable Recommendations:** the reports highlight risk factors and suggest lifestyle changes or medical follow-ups.

To integrate the LLM, a structured process was implemented:

1. **Prompt Engineering:** a detailed template was designed to summarize the SHAP analysis and prediction results, ensuring that all critical information was included and formatted for clarity;
2. **API Integration:** the LLM was accessed through the Google Generative AI API, configured with the appropriate security and performance settings;
3. **Error Handling:** mechanisms were incorporated to manage failures in text generation, including fallback strategies to notify users or request manual interpretation.

While the integration of an LLM adds significant value, it is not without challenges: LLM-generated outputs depend heavily on the quality and clarity of the input prompt. Additionally, while the LLM can provide contextually rich explanations, its outputs must be reviewed by domain experts to ensure medical validity and avoid potential inaccuracies. Ethical considerations around the confidentiality and sensitivity of health data should also be addressed, ensuring compliance with data protection standards.

In conclusion, the LLM integration bridges the gap between machine learning predictions and human decision-making by providing clear, actionable, and professional reports. This approach underscores the potential of combining predictive analytics with generative AI to improve healthcare communication and outcomes.