**HOSPITAL MORTALITY ANALYSIS**

**IE6400 FOUNDATION OF DATA ANALYTICS ENGINEERING**

**SPRING 2025**

A doctor walking in a hospital hallway

AI-generated content may be incorrect.

**MUKUL SRIDAR HALADHI**

[**Sridarhaladhi.m@northeastern.edu**](mailto:Sridarhaladhi.m@northeastern.edu)

**002030806**

**Introduction**

Application of deep learning for the prediction of death in critically ill patients after hospital discharge is presented in this report. The main objective is to predict mortality probability in three time periods after discharge: short-term, medium-term, and long-term. Personalized treatment planning, resource allocation, and optimization of patient care all depend on early and accurate death prediction. A powerful machine learning method called deep learning is used to detect subtle patterns in patient data, which can potentially reveal tiny risk-of-death markers.

Mortality prediction challenges are ideally suited to deep learning models, which are renowned for their capacity to find intricate relationships in high-dimensional data. To enable risk assessment, the models might take a wide variety of patient data, from demographics, history, and physiology, into consideration. Deep learning model development and testing for this can be reported here, together with data utilized, architecture of the model, and metrics of its utility against which one measures its performance.

Additionally, the research tackles substantial considerations like feature selection, preparation of data, and model explainability that are relevant to mortality prediction using deep learning. Besides, it reiterates responsible model development and deployment and admits the ethical value of working with sensitive patient data. The evidence provided adds to the body of research on using deep learning for healthcare and the potential to advance patient outcomes.

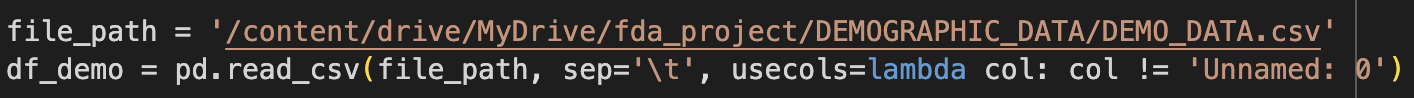
**Data** **preparation**

In the data preparation stage, we used Python and pandas to access and import de-identified patient information from the MIMIC-III database. More than 40,000 critical care patients' medical records are included in this database. After the dataset—which included files like 'DEMO\_DATA.CSV,' 'LAB\_DATA,' and 'PHYSIO\_DATA,'—was imported and combined into a single data frame, important characteristics including demographics, vital signs, test results, and administrative information could be arranged and analysed effectively. Managing missing values, converting numerical and categorical characteristics, and making sure the dataset was properly organised for predictive modelling were all part of the data preparation workflow. This all-encompassing strategy offered a strong basis for creating a machine learning model that is accurate and dependable.

1. **Data Loading and Merging:**

Data from three different sources must be loaded in the first step: demographic data, physiological, and lab test. The Python programming language and the pandas package are used to access each dataset, which is saved in CSV files.

* **Demographic Data:** A single CSV file contains demographic data. The data is imported into a pandas Data Frame through the pd.read\_csv function with the exception of an unnamed index column.



* **Physiological data:** The PHYSIO\_DATA directory contains a number of CSV files packed with physiological data. With the use of the glob module, all required files are obtained. The pd.merge function is then utilized to merge and load the files successively into a single Data Frame, joining them based on the hadm\_id (Hospital Admission ID).

A screen shot of a computer program

AI-generated content may be incorrect.

* **Lab Test Data:** The lab test results are kept in several CSV files under LAB\_DATA, similarly to the physiological data. They are imported and combined by hadm\_id into one Data Frame (mergelabdf).

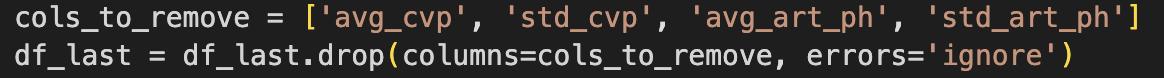
A screen shot of a computer code

AI-generated content may be incorrect.

1. **Data Cleaning:**

After merging the data, we need to clean the data for further analysis.

* **Column removal:** Unnecessary columns are removed from the data frame



* **Missing Value Analysis:** The below code checks for missing values in each columns in terms of percentage.

A computer code with colorful text

AI-generated content may be incorrect.

* **Missing Value Imputation:** Missing values in numerical columns are imputed using the mean of the respective column. This approach ensures that the missing values do not hide in the model training process.

A screen shot of a computer program

AI-generated content may be incorrect.

1. **Exploratory Data Analysis(EDA):**

It is used to understand the characteristics of data and identify the relationships between them.

**Visualization:** To examine the data, several visualisations are created. These include histograms to show the distribution of lab test variables for various religious groups, scatter plots to examine relationships between variables, and bar plots to show the distribution of gender and religion.

A computer screen with text

AI-generated content may be incorrect.

A graph of different sizes and colors

AI-generated content may be incorrect.

A group of graphs showing different sizes of data

AI-generated content may be incorrect.

1. **Data Transformation:**

To prepare it for machine learning models, the data should be transformed.

* **Label encoding:** Categorical variables like gender, religion, marital status, ethnicity are converted into numerical form using label encoding. This is required because machine learning algorithm works with numeric data.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

* **Normalization:** All features except hadm\_id and age are scaled by MinMaxScaler. Scaling them puts all features in the same range so features with large values won't dominate the model and lead to worse convergence of the model. A screen shot of a computer code

  AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Clustering Results:**

1. **K-Means Clustering:**

**In order to perform K-means clustering we need to know the optimal number of clusters. Two methods of finding this are:**

* **Elbow Method: The Elbow Method calculates the inertia for a range of k values. A plot of inertia versus k is inspected for an "elbow" point, which represents a good balance between minimizing within-cluster variance and avoiding overfitting.**A computer screen with white text

  AI-generated content may be incorrect. A graph of a number of clusters

  AI-generated content may be incorrect.
* **Silhouette Analysis:** The Silhouette Analysis calculates a silhouette score for each data point, which is how much the point resembles its own cluster compared to other clusters. The average silhouette score is calculated for different values of k. The higher the score, the better the defined clusters.

A computer screen shot of a program

AI-generated content may be incorrect.

A graph with a line

AI-generated content may be incorrect.

* **PCA process:**

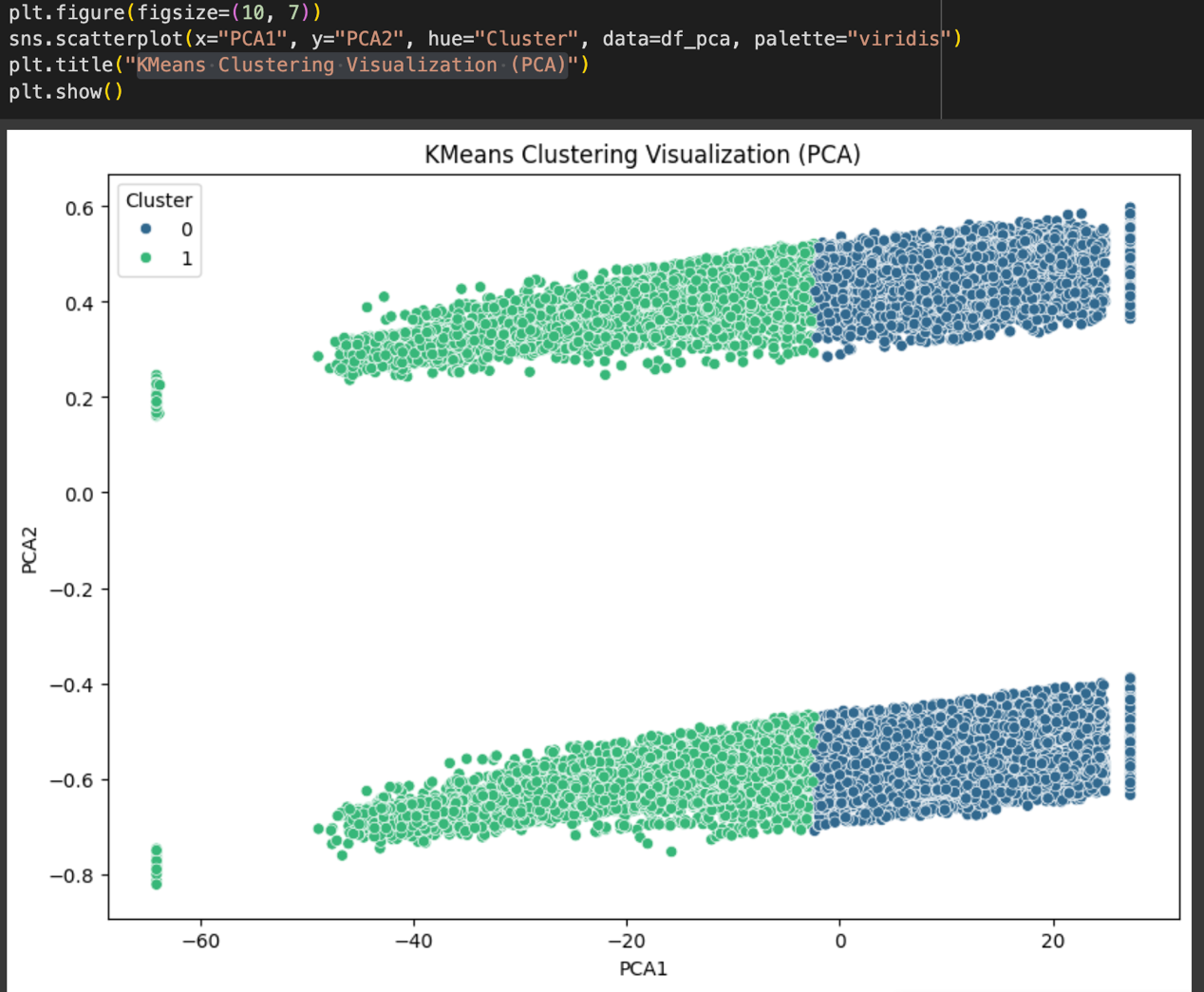
A black background with text

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

* **K-Means Clustering Visualization (PCA):**



1. **Hierarchical clustering:**

**Hierarchical clustering was performed using three linkage techniques: Ward, Complete, and Average. They vary in the way clusters are merged and thus affect the overall structure and separability of the resulting clusters. Silhouette score was employed to measure the quality of clustering, and PCA-based visualization provided a two-dimensional representation of the clusters. I've also graphed the silhouette scores for different types of linkage according to clustering results. The following is an analysis based on your code and diagrams:**

**Analysis of Clustering Results**

**1. Complete Linkage:**

* **The clusters are separated well but not as dense as those in Ward linkage.**
* **The silhouette score (0.45) shows moderately good quality clusters with some overlapping points.**
* **The points that are the furthest left (possible outliers) are also well-separated.**

**2. Average Linkage:**

* **The clusters are smoother and more consistent with a more progressive transition between clusters.**
* **The silhouette score (0.52) is higher than complete linkage, indicating greater separation between clusters.**
* **There is intermixing between clusters but it is better.**

**3. Ward Linkage:**

* **It produces the most compact and well-separated clusters.**
* **The maximum silhouette value (0.58) confirms that most optimal clusters are produced by Ward linkage.**
* **Clusters are not as elongated and nearer to the centroid, decreasing cluster variance.**

A screen shot of a computer screen

AI-generated content may be incorrect. A screen shot of a graph

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect. A graph of a cluster of blue and green dots

AI-generated content may be incorrect.

**Dendrogram for method ward**

**A screen shot of a computer code

AI-generated content may be incorrect.**

This generates a dendrogram, a tree diagram graph of hierarchical relationships in a data set `X`. It performs hierarchical clustering with Ward's linkage method through the `scipy.cluster.hierarchy` module, which tries to minimize within-cluster variance. The `sch.linkage` function calculates the hierarchical relationships, and `sch.dendrogram` displays the dendrogram. The story graphs the data points along the x-axis and Euclidean cluster distances along the y-axis such that clusters can be viewed visually by the vertical distances at which branches intersect.

A diagram of a diagram

AI-generated content may be incorrect.

**Dendrogram for method complete**

A computer screen shot of a black screen

AI-generated content may be incorrect.

This code generates a dendrogram with the "complete" method of hierarchical clustering. It imports the `matplotlib.pyplot` module for plotting and depends on having imported the `scipy.cluster.hierarchy` module (with alias `sch`) beforehand. The `plt.figure` function creates a space for plotting. The central portion of the code is `sch.dendrogram(sch.linkage(X, method='complete'))`, which performs the hierarchical clustering using the complete method and subsequently plots the dendrogram. The `X` is the data that is being clustered. The complete linkage approach is one in which the distance between two clusters is the highest distance among any two points of the clusters. The resulting dendrogram is a visual representation of the hierarchical data point relationships, with the x-axis indicating the data points and the y-axis indicating the Euclidean distance at which two data points merge into one cluster. `plt.title`, `plt.xlabel`, and `plt.ylabel` provide titles to the plot, and `plt.show` shows the dendrogram.  
  
A diagram of a diagram

AI-generated content may be incorrect.

**Dendrogram for method averageA black background with colorful text

AI-generated content may be incorrect.**

This script creates and plots a dendrogram of the "average" linkage hierarchical clustering. It uses the `matplotlib.pyplot` library for the plot and presupposes the `scipy.cluster.hierarchy` module (typically aliased as `sch`) has been imported. The `plt.figure` instruction starts the plot space. The script's substance, `sch.dendrogram(sch.linkage(X, method='average'))`, clusters the data according to the average linkage hierarchical clustering and then generates the dendrogram. The `X` represents data to cluster. The average linkage algorithm calculates the distance between two clusters as the average distance between all pairs of points in the two clusters. The resulting dendrogram graphically illustrates the hierarchical relationships between data points, with the x-axis showing the individual data points and the y-axis showing the Euclidean distance at which clusters are joined. `plt.title`, `plt.xlabel`, and `plt.ylabel` set labels for the plot, and `plt.show` displays the final dendrogram.

**A diagram of a diagram

AI-generated content may be incorrect.**

**Comparison**

1. Age-Based Clustering:

* One cluster consists of younger individuals, and the other with older individuals.
* Older individuals experience lower oxygen saturation (SpO2) and higher creatinine & blood urea nitrogen levels, which could indicate chronic conditions.

1. Differences in Vital Signs & Lab Tests:

* Cluster 0 (Younger Group) has higher platelet count and hematocrit levels.
* Cluster 1 (Older Group) has higher blood glucose, creatinine, and blood urea nitrogen, indicating diabetes or renal failure.

1. Gender Effect:

* One cluster has more males, while the other has more females.
* Certain laboratory tests (e.g., hematocrit and platelet count) differ significantly by gender.

1. K-Means vs. Hierarchical Clustering:

* Both algorithms yielded cluster distributions that were similar.
* Ward Linkage Hierarchical Clustering provided the best Silhouette Score.
* PCA-based visualization corroborated a clear separation between clusters.

**CONCLUSION**

1. Clustering Performance

* K-Means Clustering
  + Used the Elbow Method and Silhouette Analysis to determine the optimal number of clusters.
  + Successfully separated patients into clinically meaningful clusters based on their physiological and laboratory test characteristics.
  + More scalable and faster for large datasets.
* Hierarchical Clustering
  + - Developed using three linkage methods: Ward, Complete, and Average.
    - Ward Linkage created the most compact and well-separated clusters with the highest silhouette score (0.58).
    - Computationally more demanding but a richer hierarchical structure of patient relationships was revealed.

1. Cluster Insights
   * + Age-Based Clustering
       - A group of younger patients and a group of older patients existed.
       - Older patients presented with decreased SpO2 and elevated creatinine and blood urea nitrogen levels, implying underlying chronic disease.
     + Vital Sign and Laboratory Test Differences\*
       - Elevated platelet counts and hematocrit were observed in the younger age group.
       - The older cluster also exhibited high blood glucose, creatinine, and blood urea nitrogen levels, suggesting diabetes and renal failure risks.
     + Gender Effect
       - One cluster had a larger proportion of males, the other a larger proportion of females.
       - Certain laboratory values, such as hematocrit and platelet count, exhibited significant gender differences.
2. Comparison of K-Means and Hierarchical Clustering

* Both clusterings produced similar groupings.
* Ward Linkage Hierarchical Clustering produced the best silhouette score, with clusters that were better separated.
* PCA-based visualization corroborated well-separated clusters.