CS7641 Machine Learning Group 21

View the Project on GitHub TJeng7/CS7641-group-21

Introduction and Background

With the advent of modern technology and urbanization, societies have gradually cultivated lifestyles which inadvertently increase the risk of individuals developing cardiovascular diseases. Our model aims to detect at-risk individuals by utilizing a dataset sampling males in a heart-disease high-risk region of the Western Cape, South Africa to identify trends and predict at-risk individuals which will fuel timely medical interventions and guide our future healthcare decisions. Further information available at https://www.kaggle.com/datasets/yassinehamdaoui1/cardiovascular-disease. After implementing a few preprocessing models and machine learning models, we realized that this dataset did not have enough samples for a suitable solution. Thus, we changed to the following dataset which includes 70,000 datapoints: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.

Problem Definition

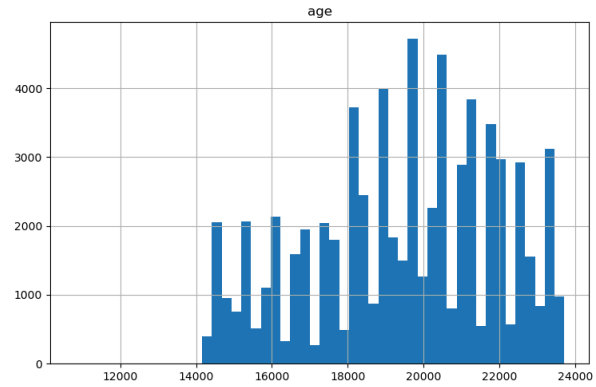
Globally, cardiovascular diseases remain a persistent threat, claiming more than 17.3 million lives every year; this is more than tuberculosis, HIV, and malaria combined [1]. Though numerous studies have tried to quantify and control these conditions, new issues arise from the research: medical data tends to reside in fragmented systems, making it quite difficult to aggregate the data for proper analysis. In addition, previous studies have lacked adaptability to individuals and regions on top of the vast computational complexity of analyzing swaths of medical data. We propose a machine-learning algorithm to highlight key trends in our dataset to simplify the process of assembling data and converting it to actionable knowledge in a scalable, efficient manner.

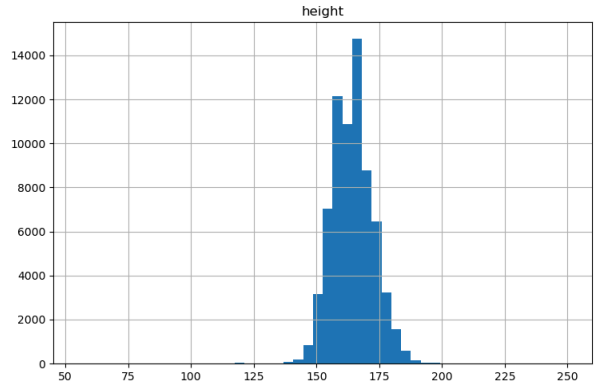
Methods

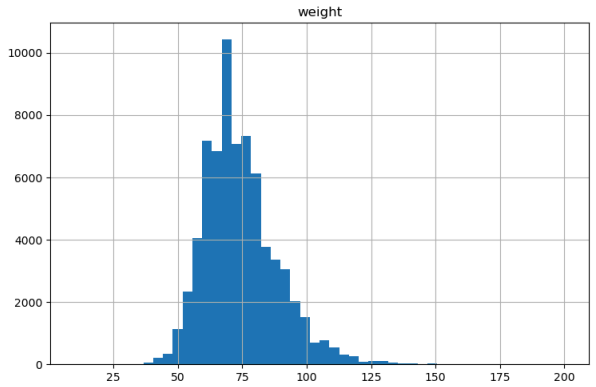
Data Preprocessing Methods:

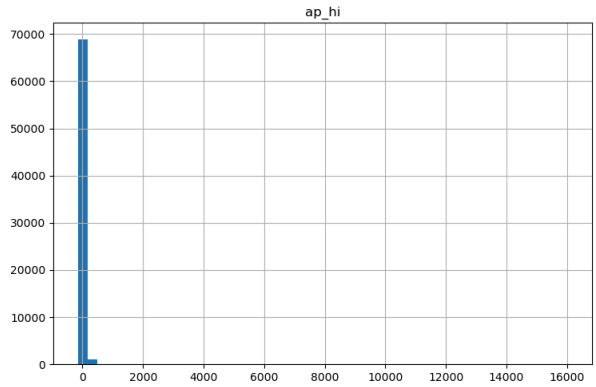
Outlier Removal and Dataset Reasonableness: There are some datapoints with very high blood pressures, with the systolic blood pressure sometimes as high as 10k+. Of course, no human can have that high of a blood pressure, so datapoints with blood

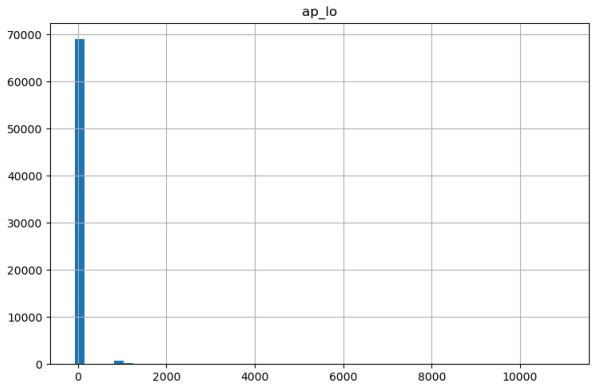
presssure readings below 250 were only considered. Also, negative blood pressure values were filtered out. To filter out outliers, every continuous feature was looked at and datapoints with feature(s) that were three standard deviations away were filtered out. Below shows the initial distributions of each continuous feature.









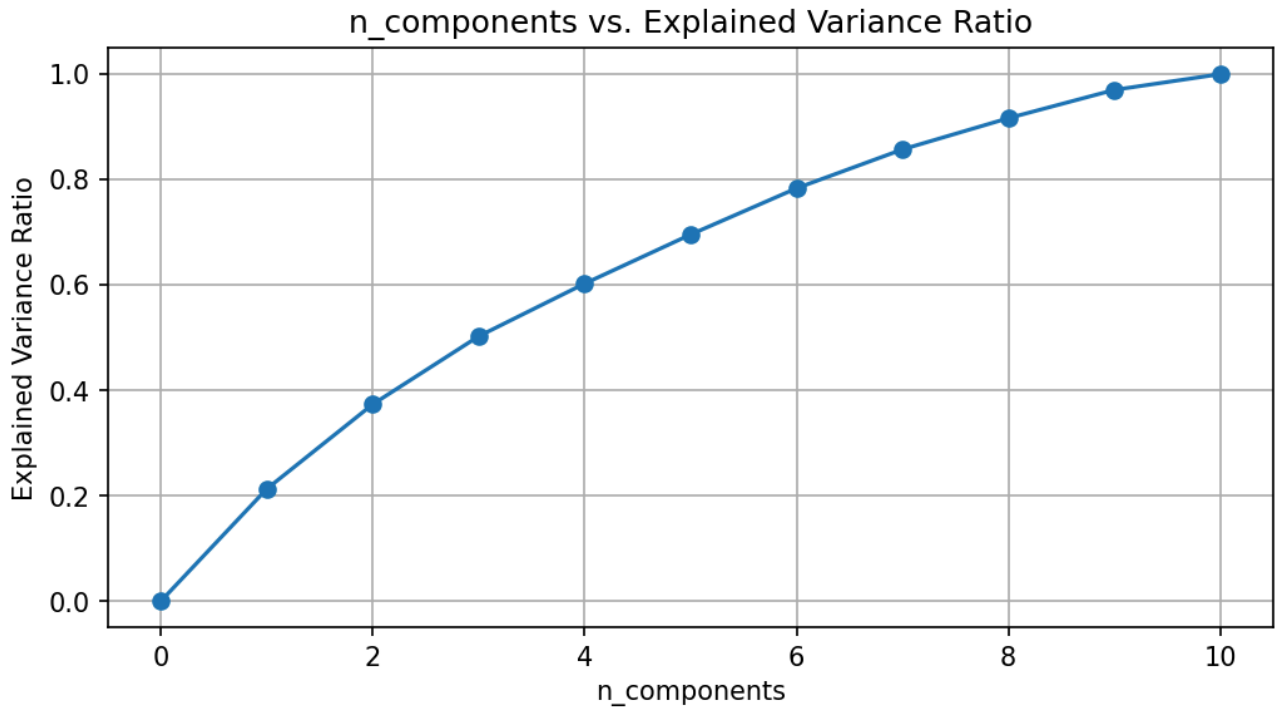


Standardization: In order to prepare our values for use with the ML algorithms, we use Z-score standardization on our dataset. This preserves the magnitude of the data while allowing us to compare the values using the same scale.

Feature Combining: To further our preprocessing on the data, we looked for a way to reduce the number of features by combining columns. We saw that the height and weight columns could be combined into BMI, so feature combining is a preprocessing method that takes the height and weight columns and converts the data into BMI once this is calculated. The new BMI column is added to the dataset (the height and

weight columns are removed). This allows us to reduce the number of dimensions (simplifying the data) and increase efficiency in future ML algorithms.

Principal Component Analysis (PCA): Our final preprocessing step was to perform PCA on the data, further reducing the number of features. We chose PCA in order to capture the features associated with the most variance, helping reduce the risk of overfitting and the computational time needed to run our machine learning algorithms. Additionally, when running PCA on our dataset, we were able to reduce the number of features from 10 to 8 while maintaining a variance ratio of > 90%. Below depicts the explained variance by the number of principal components.



Machine Learning Algorithms/Models:

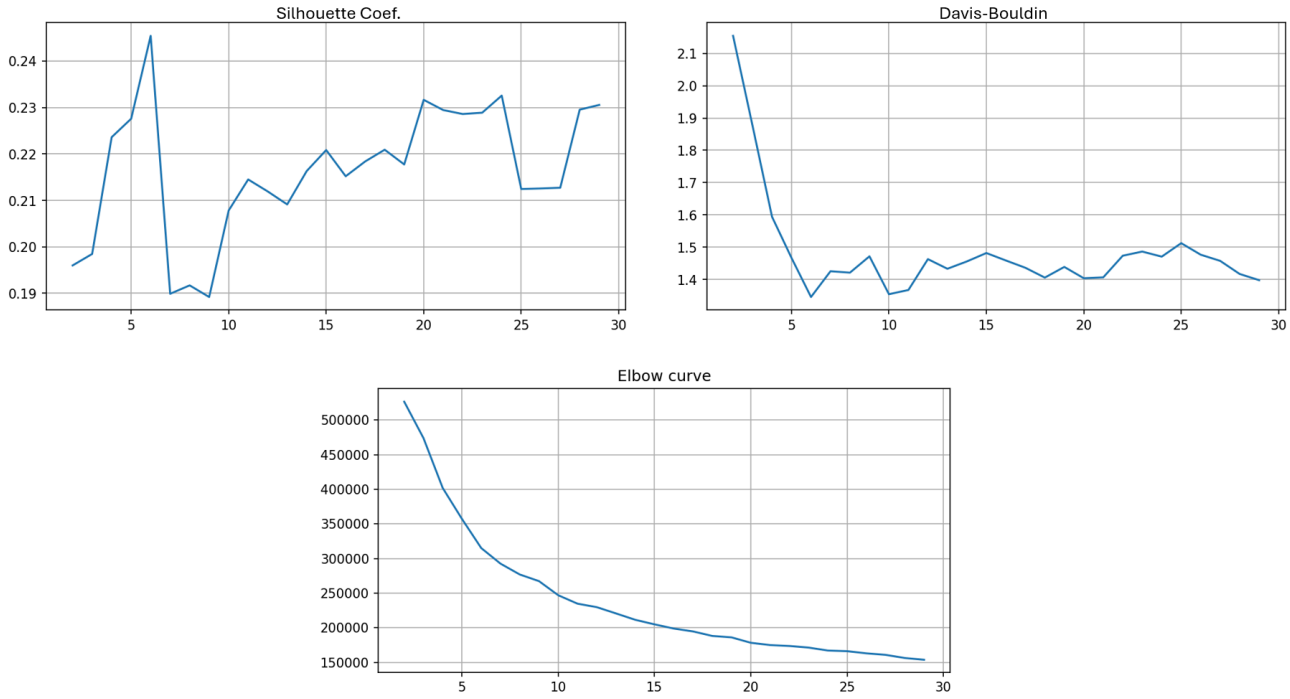
For our ML algorithms, we implemented K-means, GMM, Logistic Regression, and Random Forest. Using our dataset and Scikit-learn's corresponding libraries (Kmeans and GaussianMixture), K-means and GMM enable us to group patients based on similarities in their health metrics, allowing us to identify groups at risk of cardiovascular disease. For Logistic Regression, it assumes a generalized linear relationship between the features and the target variable (cardiac failure risk), which makes it useful for risk assessment, leading us to use the Scikit-learn's LogisticRegression library [3]. Finally, Random Forest is a supervised algorithm used for making predictions [4]. Using our dataset and Scikit-learn's RandomForestClassifier library, it can help classify a patient based on different health metrics and predict a patient's risk of developing heart disease.

Results and Discussion:

At the project's end, we implemented models that accurately reflect the diagnosis of a patient being tested for cardiovascular disease based on test results and day to day activities. The metrics used to quantify this are the accuracy and F1 score (precision and recall). Since we want to minimize the false negatives, the Recall rate should be high, so we are targeting around 75%. And finally, to get a holistic study, we utilize the F1 score which considers precision and recall to study the quality of the health model. The goal for this is to have an accuracy score that is greater than 75% [5]. Below shows our completed implementations:

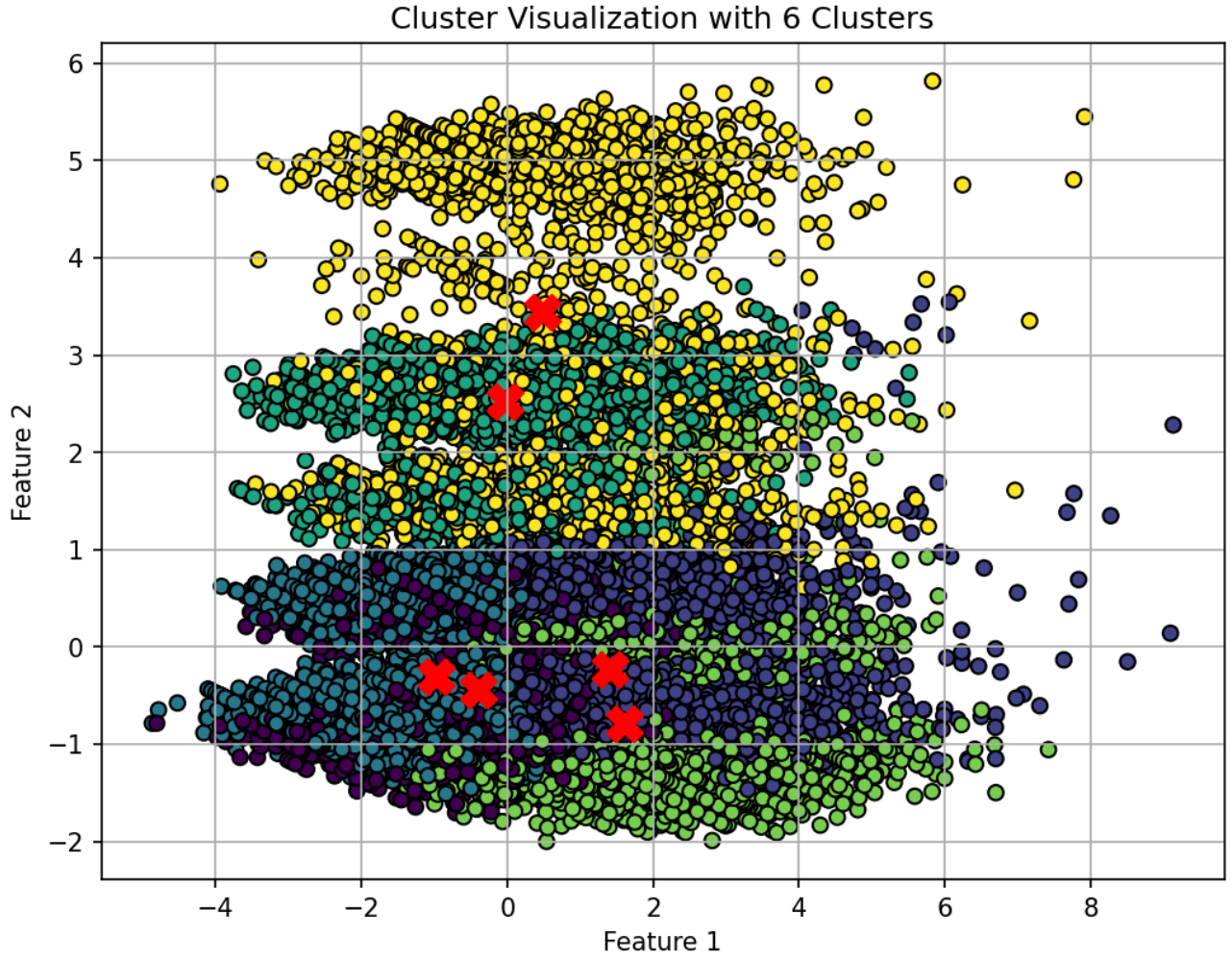
Results and Discussion - Algorithms

K-means: For our first ML algorithm, we implemented K-means, in hopes of finding trends within similar groups of data. After running K-means on our processed data, we plotted the following graphs to evaluate how many clusers to use, and run k-means clustering on our data:



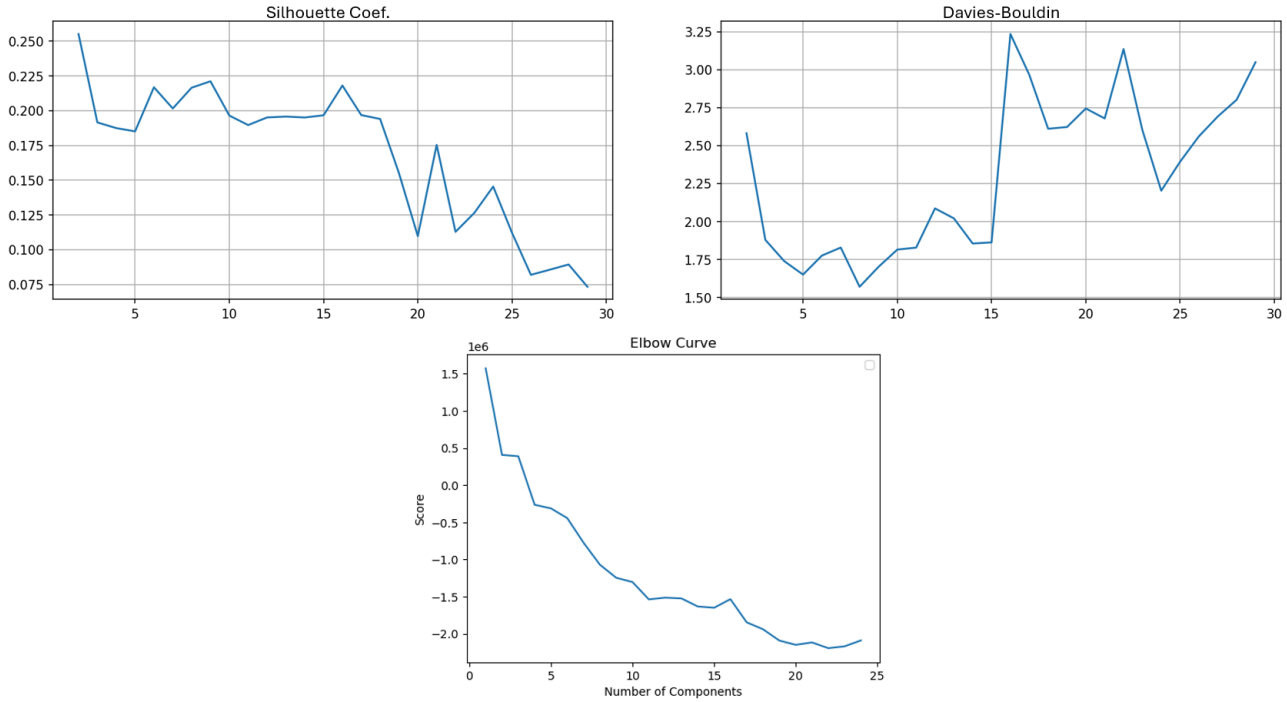
Looking to the graph above, we see the elbow curve suggests that the optimal number of clusters for this dataset is 6 clusters. This was indicated as a clear elbow on the graph occurs at the point of 6 clusters. This measure balances cluster compactness while minimizing the number of clusters, however it doesn't guarantee optimal clusters. While optimal number of clusters is not guaranteed by the elbow graph, the choice of 6 clusters is further confirmed by the silhouette analysis, which outputs scores ranging between -1 and 1. Running a silhouette analysis on our data we reached a max at 6 clusters with a score of 0.24541. This low score of 0.24541 implies that the clusters are not very well-separated and further are weakly

correlated, something that is mirrored in the k-means visualization. Upon running the Davies-Bouldin score, where lower values are preferred, the lowest score of 1.34553 was achieved with 6 clusters. Since all three evaluations point towards 6 being the optimal number of clusters, we selected to use 6 clusters. Our clustering according to 6 clusters is shown below:

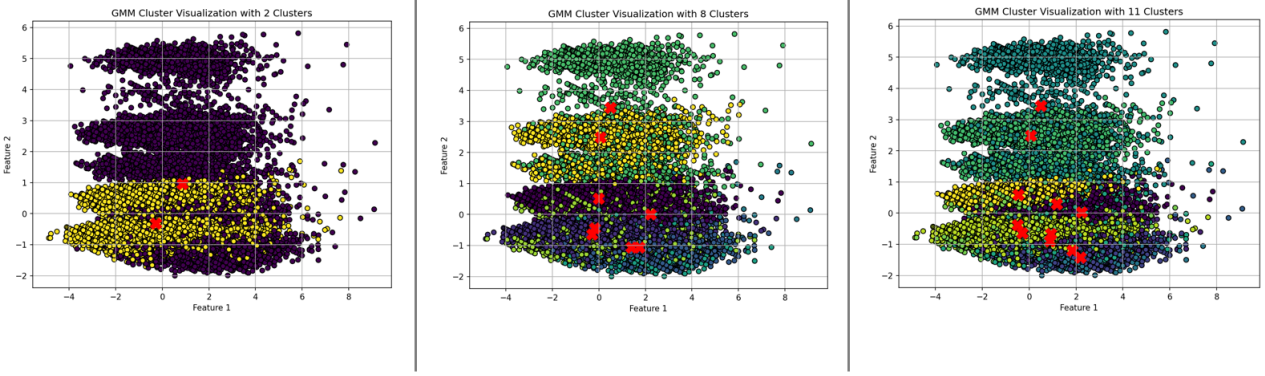


Upon analysis of the cluster visualization, we see 5 clear clusters to the human eye, however they were not properly clustered by k-means. These clusters are stretched and rotated, despite being clear cut, which is why k-means, simply clustering based upon Euclidean distance, does not seem to be the optimal analysis algorithm to run on our data. Along with this, our chosen clusters seem to be un-optimal as they are mostly concentated within one key cluster. This is most likely due to a density situtation where one cluster of data contains most of the data, throwing off the k-means clustering eval. Moving forward, we will account for this density which is throwing off k-means. As the cluster plot with 6 clusters illustrates poorly separated clusters, suggesting that they do not effectively capture distinct sub-groups within the data, it is further confirmed that k-means is not the optimal algorithm for our data. Moving forward, we thus intend to look to a different clustering algorithm that accounts for stretching and rotation.

Gaussian Mixture Model: Due to the stretched and rotated nature of the clusters, we then decided to implement GMM as we believed this algorithm would better fit the data. We first decided to find the optimal number of clusters to use by running GMM using 2 to 30 clusters. Below shows graphs depicting the three different clustering evaluations (Elbow Curve, Silhouette Coefficient, and the Davies-Bouldin index) that we used:



For our elbow curve, we used BIC to identify the optimal number of clusters. Looking at the graph, there seems to be an elbow at 11 clusters. However, based on the Silhoutte Coefficient and Davies-Bouldin index, the optimal number of clusters is 2 and 8, respectively. According to the Silhouette Coefficient, a cluster number of 2 yielded a low score of 0.25512. Similar to our Silhouette coefficient score in K-means, this low value indicates relatively weak clustering and may indicate that the data is not well suited for GMM. Further reinforcing this, when looking at the Davies-Bouldin evaluation, the optimal cluster number of 8 provided a high score of 1.57025, indicating relatively poorly separated clusters and weak clustering. Since each of the three clustering evaluation methods gave different values for the optimal number of clusters, we decided to run GMM with each of the three clustering values. Below shows our clustering results:



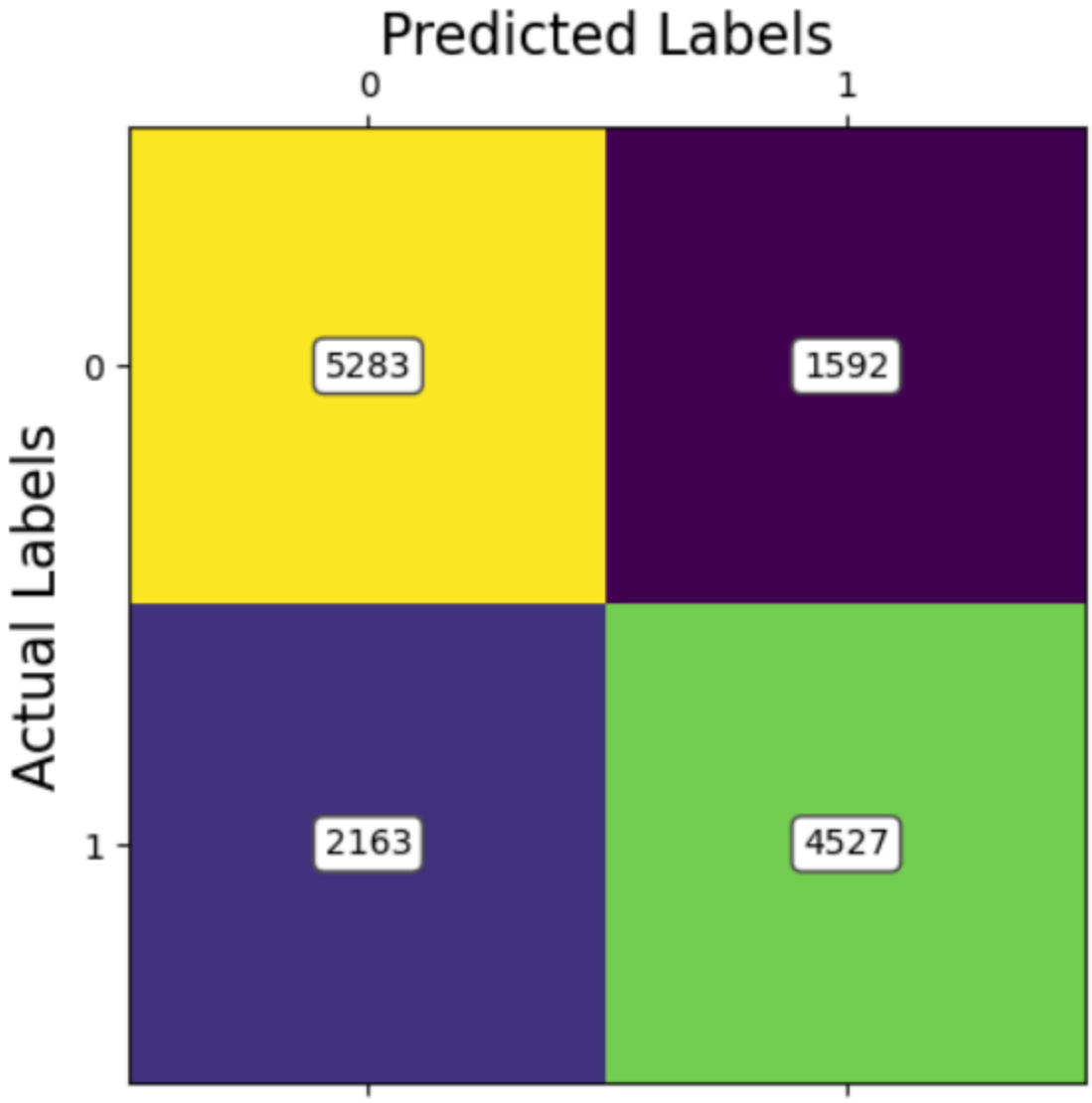
Mirroring K-means, there appears to be 5 clear clusters visible in the graph that are not properly clustered by GMM. Despite data appearing to be well suited for GMM due to the stretched nature of the visible clusters, GMM does not appear to be able to pick up on them and GMM's clustering appears weak as reinforced by our clustering evaluation scores.

Initially, we believed that GMM would perform well with our data as the visible clusters appear to be stretched and seem to follow a distribution close to Gaussian. However, we believe that GMM may not have been suited to our data since it is likely that the data points in a cluster do not follow a Gaussian distribution. When graphing the clusters, we chose the two most informative features (features that maintain the highest variance) for plotting the x-axis and y-axis in order to visualize the clustering in 2D. However, it is likely that these two features are not fully representative of our data and while it appears that the data points in a cluster follow a Gaussian distribution as seen in our visualization in 2D, this is likely not the case when considering all of our features. As such, a different clustering algorithm may be better suited for our data.

When comparing our results to K-means, we do not notice improved clustering with GMM. Specifically, GMM's Silhouette Coefficient score is slightly higher than K-means (GMM: 0.25512 vs K-means: 0.24541), which demonstrates better clustering. However, K-means' Davies-Bouldin index is lower than GMM's (GMM: 1.57025 vs K-means: 1.34553), indicating better clustering on K-means' end. Additionally, when looking at the clustering graphs, GMM does not appear to be better suited as it does not properly capture the 5 visible clusters, similar to our results from K-means.

Logistic Regression: We then implemented logistic regression as a supervised model for our data. Considering we are focused on predicting a binary output (0 for non-risk and 1 for at-risk), we hoped this would be an optimal choice. The resulting confusion matrix, shown below, displays an accuracy of 72.32%, precision of 73.99%, and recall of 67.67%. The F1 score is 70.68%. These values provide a wholistic view of the

performance of our model, showing that our model has a significant advantage over random assignment.



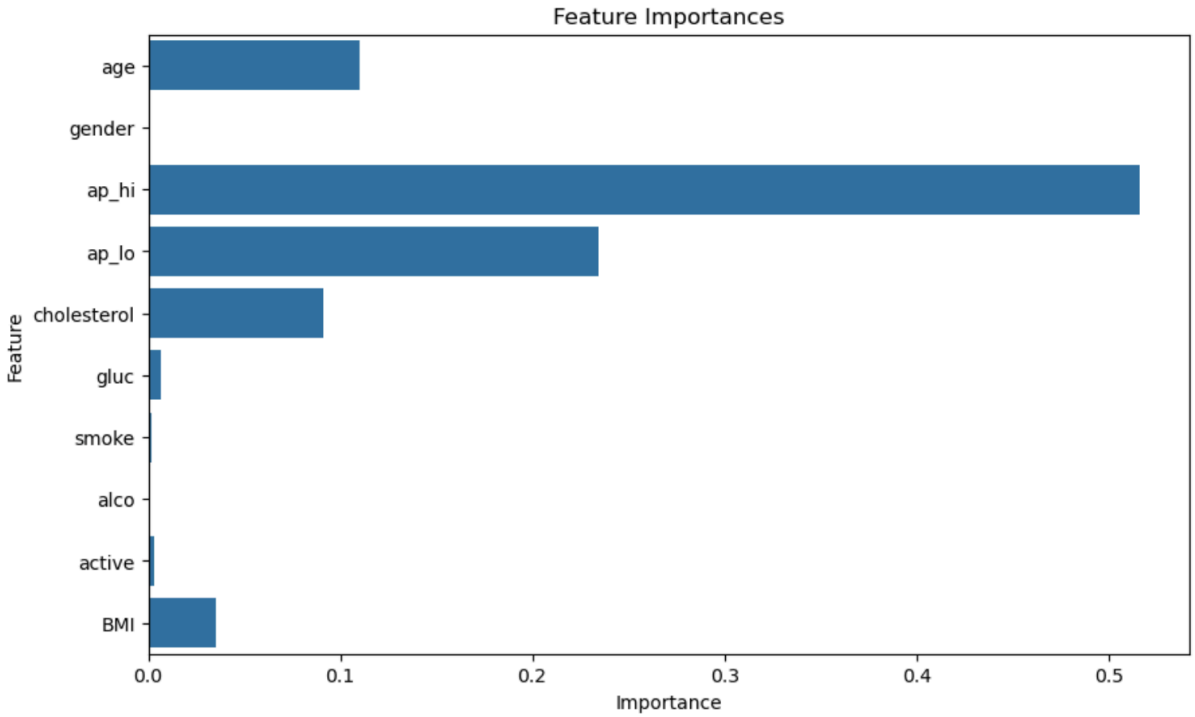
Though we tested multiple threshold values such as 0.3 and 0.7 for our logistic regression implementation, the default value of 0.5 resulted in the highest accuracy. Additionally, we optimally subsampled our dataset using a testing/training set split of 20/80% test data to ensure that we preserved the set's variability while not overfitting the training data. Using kfold cross validation, we confirmed the consistency of our predictions. We used 5 splits for this verification, resulting in an accuracy of 72.14%. Comparing this to our base model's 72.32%, we can corroborate the model's performance. Of course, setting the threshold lower will reduce the number of false negatives, which may be good since telling someone they do not have a disease when they actually have it could be the difference between life and death. Setting the threshold to 0.3 will increase recall to about 90% but the tradeoff is it would drop the accuracy to 64%.

We also attempted to utilize forward selection to iteratively maximize our model's predictive performance; we selected the features 'age', 'ap_hi', 'cholesterol', 'smoke', and 'alco', which correspond to the individual's age, arterial pressure hi (blood pressure value), cholesterol levels (in terms of average, above average, or significantly above average), smoking habits (binary), and alcohol consumption habits (binary). However, forward feature selection did not significantly change the f measures, resulting in a marginally increased metrics including an accuracy of 72.61%, precision of 75.37%, recall of 66.04%, and F1 score of 70.4%.

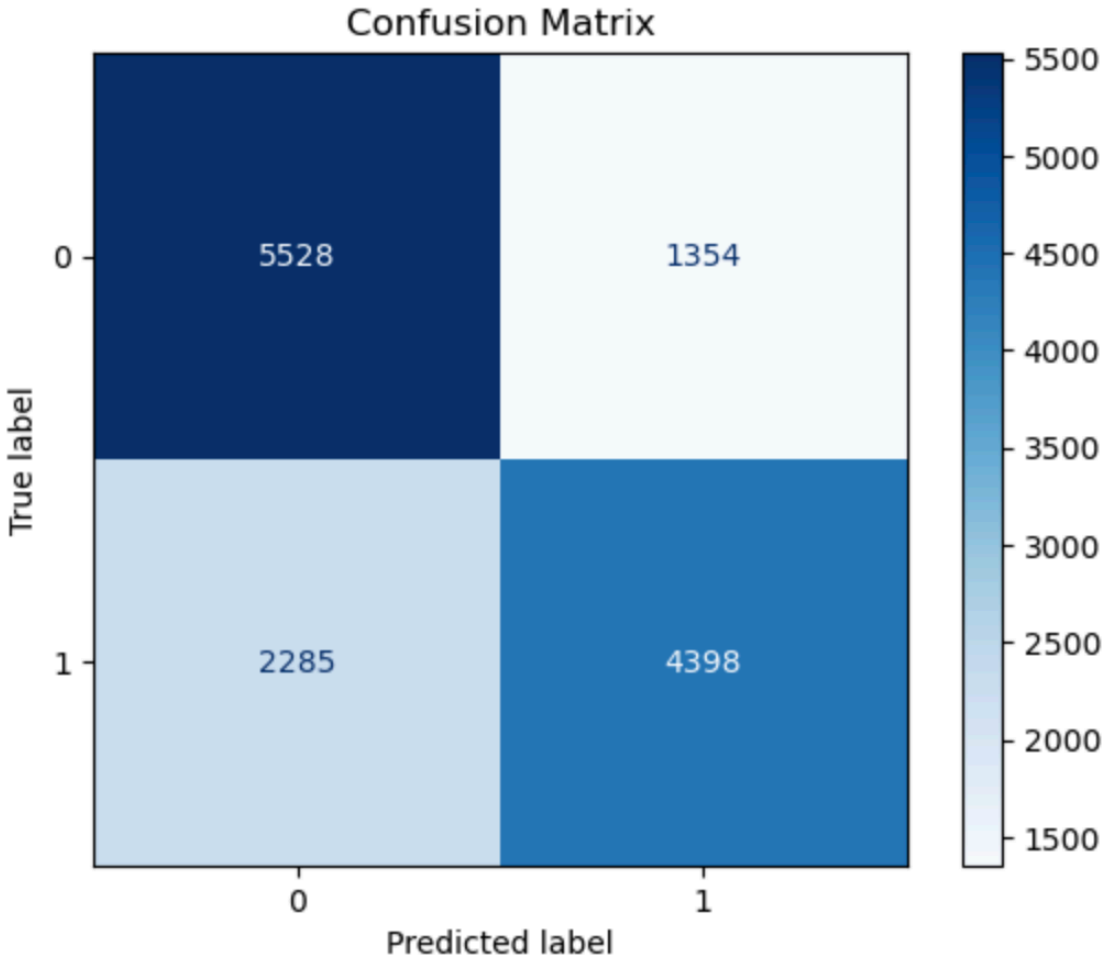
Though we did not achieve our target accuracy and F1 measures of ~75% through logistic regression, our model serves as a substantial improvement over random designation.

Random Forest: We additionally chose to implement random forest as a supervised model for our data. Random forest allows us to understand the relationships between different features and which ones are more useful than others. Once again as we are dealing with a binary classification model to determine: 0 for non-risk of cardiovascular disease and 1 for at-risk, we believed this would be an optimal choice that included a different approach from logistic regression.

As mentioned earlier, random forest allowed us to understand the complex relationships between features to train the model. By plotting feature importances, it is clear that blood pressure values are a very important measure for cardiovascular disease. This conclusion was expected. An important note/flaw though is that activeness, smoke, and alcohol are shown of least importance. Because the dataset encoded binary values for smoking, alcohol, and activity it was difficult to use their values to determine the result. There was not a range for how much someone does of the particular activity. In the real world, there is a range for all these activities and because of that they do have a lot of importance when concluding whether someone may have a cardiovascular disease or not.



In a general review of our random forest model, we saw a reasonable classification performance of the model which can be seen in the model predicting risk for cardiovascular disease risk in males and achieving an accuracy of 73.2%. Further looking into these results we have a confusion matrix (depicted below) to visualize our data and the classification performance of the model. The confusion matrix values show 5,528 true negatives, 1,404 false positives, 2,180 false negatives, and 4,398 true positives. While we do see that our model generally exhibits the correct classification of most cases, it seems to struggle specifically with misclassifications being false negatives. This is not ideal for a medical model as it leaves patients at risk of cardiovascular disease under the impression that treatment or a life style change is not necessary. This specifically is most likely people who are at a lower-risk of cardiovascular disease. This is a challenge in medical contexts, as underestimating risk may delay critical interventions. Further moving forward, we would aim to reduce false negatives at the risk of increasing false positives and model performance.



Additionally, we tuned the hyperparameters to avoid under-fitting/overfitting. The 'max_depth' hyperparameter focused on the fitting because larger depth meant a chance for overfitting whereas a lower depth could lead to under-fitting. The 'n_estimators' hyperparameter determined how many trees would be created and helped to find a balance between accurate representations and the efficiency of the algorithm. Since random forest is expensive for computations, the tuning found the correct balance. Looking into the tuned parameters of 'max_depth'=6 and 'n_estimators'=163, we see values that suggest a balance between model complexity and overfitting. Taking into account the also larger value of false positives we see in our data this indicates potential over-sensitivity in the model. While this model is performing better than average, there are clear areas for improvement.

From running Random Forest, we got an accuracy of ~73.2%, precision of ~76.4%, recall of ~65.8%, and F1 of ~70.7%. While we did not achieve our target values for accuracy and recall, there seems to be some improvement over logistic regression in accuracy, precision, and F1 score. In conclusion for Random Forest, it is generally able to exhibit the correct classification and provide improvement in certain results over

logistic regression. There are also limitations that occur because of the dataset and the encoding of values for certain features.

Comparison of Algorithms Performance

K-Means versus Gaussian Mixture Model: First diving into how our unsupervised models compared, we see that both K-means and GMM struggled to effectively cluster the dataset. We assume this is likely due to the data's density, which highly threw off k-means, and non-Gaussian distribution of the data. K-means achieved a slightly better Davies-Bouldin score of 1.34553 versus GMM's 1.57025. This indicates better compactness and separation in K-means. However, k-means' Silhouette Coefficient score of 0.24541 was slightly lower than GMM's of 0.25512, which reflects a weak overall clustering for both algorithms on the data set. GMM was expected to handle the stretched clusters better but failed due to our assumption of a Gaussian distribution in our dataset, which likely doesn't hold across all features. Neither method captured the visually apparent 5 clusters, suggesting that the dataset's structure may require a density-based algorithm like DBSCAN or additional feature engineering for better performance.

Logistic Regression versus Random Forest: For the supervised models, we have logistic regression and random forest. Comparing results, we can see that there are similar values among the two models. As mentioned earlier, logistic regression has an accuracy of 72.6%, precision of 75.4%, recall of 66%, and F1 score of 70.4%. Random forest has an accuracy of 73.2%, precision of 76.4%, recall of 65.8%, and F1 of 70.7%. Random forest has a small improvement in accuracy, precision, and F1 score over logistic regression whereas the recall score in logistic regression is better. It makes sense that random forest has better accuracy since it is able to represent the complex relationships between features whereas logistic regression uses a linear model. There are many features within this dataset (blood pressure, cholesterol, alcohol, smoking, etc.) which lead to a more complex model hence meaning that random forest may be able to classify results better. Logistic regression has a slightly higher recall score which means that it is better at finding people with cardiovascular disease, but since its precision was lower it means that it had more false positives (logistic regression seems to be classifying many of the datapoints as having CVD). Both the models have comparable F1 scores meaning they both balanced the precision and recall similarly. Logistic regression and random forest classification model the data differently and hence they gave different results in terms of accuracy, recall, and precision, but overall they seemed to perform about the same (F1 score), with random forest classification doing slightly better in certain scenarios.

Next Steps

For the future, our next step would be to implement DBSCAN as our two clustering algorithms (K-means and GMM) performed relatively poorly. We believe that DBSCAN can perform well due to the density of our data, as indicated in earlier sections. Additionally, another step would be to further tune our hyper-parameters in order to better improve our algorithms performance.

References

[1] “CARDIOVASCULAR DISEASE STATISTICS REFERENCE DOCUMENT.” Available: https://www.heartfoundation.co.za/wp-content/uploads/2017/10/CVD-Stats-Reference-Document-2016-FOR-MEDIA-1.pdf

[2] K. Mo, “Hands-On PCA Data Preprocessing Series. Part I: Scaling Transformers,” Medium, Jun. 11, 2020. https://towardsdatascience.com/pca-a-practical-journey-preprocessing-encoding-and-inspiring-applications-64371cb134a

[3] IBM, “What Is Logistic Regression?,” IBM, 2024. https://www.ibm.com/topics/logistic-regression

[4] IBM, “What is a Decision Tree?,” IBM, 2023. https://www.ibm.com/topics/decision-trees

[5] N. Sharma, “Understanding and Applying F1 Score: A Deep Dive with Hands-On Coding,” Arize AI, Jun. 06, 2023. https://arize.com/blog-course/f1-score/#:~:text=F1%20score%20is%20a%20measure

Project Contributions

Name	Project Contributions
Andy Kao	PCA Implementation/Analysis + Clustering Evaluations + GMM Implementation/Analysis
Giselle McPhilliamy	Random Forest Implementation + relevant visualization and statistics ; Algorithm Comparisons
Jaden Lim	Logistic Regression script + relevant visualizations and statistics
Ramya Subramanian	Random Forest Implementation + relevant visualization and statistics ; Algorithm Comparisons
Tyler Jeng	Logistic Regression script + relevant visualizations and statistics

Gantt Chart (use GT affiliated email):

Gantt Chart

Github Repository

/data/ : Directory containing our datasets.
/data/cardio_train.csv : Updated dataset with 70,000 datapoints.
/data/cardiovascular.txt : Old dataset. Did not perform well with our model, thus it was swapped out.
/algorithms/ : Directory containing supervised and unsupervised algorithms.
/algorithms/kmeans.py : File containing cluster evaluations and k-means clustering algorithm.
/algorithms/gmm.ipynb : Notebook containing cluster evaluations and GMM clustering algorithm with relevant visualizaations.
/algorithms/algorithms.ipynb : Notebook containing preprocessing methods and algorithms (K-means and Random Forest) for easier visualization.
/algorithms/log_reg.ipynb : Notebook with scripts for performing logistic regression (with cross validation and forward feature selection) with relevant visualizations
/images/ : Directory containing images for the report.
/visualizations/ : Directory containing visualizations.
/visualizations/eda.ipynb : Notebook containing visualization for exploratory data analysis.
/preprocessing scripts/ : Directory containing scripts which clean and preprocess the dataset.
/preprocessing scripts/standardize.py : Preprocessing script for standardizing the dataset using Z-scores.
/preprocessing scripts/removeoutliers.py : Preprocessing script for removing outliers from the dataset.
/preprocessing scripts/featurecombining.py : Preprocessing script for combining height and weight features into BMI.
/preprocessing scripts/pca.py : Preprocessing script for performing PCA on the dataset.

This project is maintained by Tyler Jeng, Andy Kao, Giselle McPhilliamy, Ramya Subramanian, and Jaden Lim .