**Advanced Data Mining for Data-Driven Insights and Predictive Modeling**

Rutu Shah

Nischal Joshi

Murali Krishna Chintha

Santhosh

University of the Cumberlands

MSCS-633 Advanced Big Data and Data Mining

Dr. Satish Penmatsa

07/13/2025

**Abstract**

In this project of predictive modeling, we have demonstrated an application of advanced data mining techniques on a medical dataset related to heart disease. The complete data science pipeline is covered, including data preprocessing, exploratory data analysis (EDA), feature engineering, regression and classification modeling, clustering, and association rule mining. Each phase is analyzed for its contributions to deriving clinically relevant insights and improving prediction accuracy. In addition to this, we have also considered ethical implications that surround model bias, data fairness, and privacy. The result of this depicted the value of machine learning in the predictive healthcare system and established a solid foundation for future research or real-world deployment.

*Keywords: Data pre-processing, Feature Engineering, Data Cleaning, Clustering, Classification*

**Introduction**

Heart disease is one of the major health issues globally, contributing significantly to mortality and morbidity. To improve the health of patients, it is essential to do early detection and optimize healthcare resources. The main agenda of this project is to apply data mining techniques to identify whether predictive analysis and patterns associated with heart disease are helpful in evaluating multiple modeling approaches for accuracy that are clinically applicable worldwide. In this project, we covered both supervised and unsupervised learning algorithms, which predict outcomes and discover the new latent patterns in the data (Friedman et al., 2001).

**Dataset Description**

For this task we have selected the "Heart Disease UCI" dataset obtained from Kaggle, which includes data from multiple sources/locations such as, Cleveland, Hungary, Switzerland, VA Long Beach. It includes 920 patient records with 16 attributes. The data set consists of medical attributes gathered from patients in order to forecast the severity of heart disease. The attributes are: age, sex, chest pain type (cp), blood pressure (trestbps), cholesterol(chol), fasting blood sugar (fbs), resting ECG results (restecg), maximal heart rate achieved (thalach), exercise-induced angina (exang), ST depression (old peak), slope of the peak exercise (slope), major number of blood vessels (ca) and thalassemia (thal), predicated attribute(num).

The dataset was selected because of its real-world relevance as well as diversity of features. Having sufficient sample records is essential to apply and compare different machine learning models (Kaggle, n.d.). Once the data was selected we carried out data collection, cleaning and exploration. For data cleaning the data was first explored using in-built methods of python and after getting insight of the initial data the duplicate and unnecessary values were removed. The missing values are diagnosed using df.isnull().sum. The count of the missing values along with their corresponding attribute are as follows:

trestbps : 59

chol: 30

fbs: 90

restecg: 2

thalach: 55

exang: 55

oldpeak: 62

slope: 309

ca: 611

thal: 486

Similarly, the duplicate values were detected using df.duplicated() and removed using df.drop\_duplicates(). For data exploration we used df.info() and df.astype(). As for the outliers and noises, the visualization distribution was carried out using histogram and boxplots which helped in identification and demonstration of outliers and noises.

**Challenges Faced:**

Selecting the appropriate dataset from the kaggle was the major challenge.Initially we worked in a titanic dataset however it lacked the numerical attributes which is essential for further computation. Later we worked with a heart disease data set with a lot of numerical values .

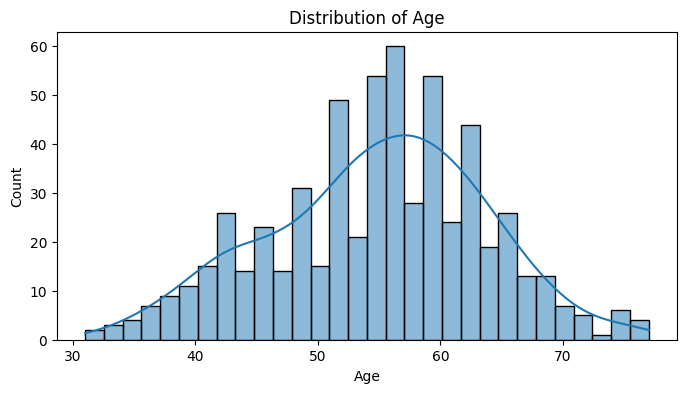
Several columns in the dataset (e.g., trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal) contained missing values, some with very high missing rates (e.g., ca, thal, slope). This posed a risk of losing too much data if all rows with missing values were dropped.

Lastly, outliers and unrealistic values present in the numeric columns (age, chol) could bias the models and reduce performance.

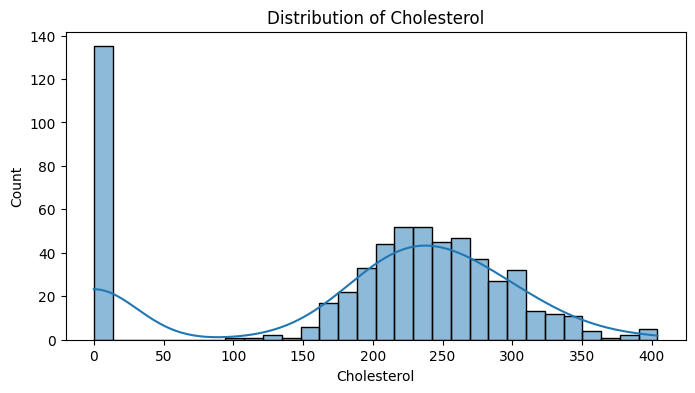
**Exploratory Data Analysis (EDA)**

The high number of heart disease patients are between the age of 40 and 60 where men comprised a larger portion of the dataset and showed higher prevalence of heart disease. Additionally, the patients with high blood pressure and low heart rate had a greater likelihood of having heart disease.For visualization following steps were taken:

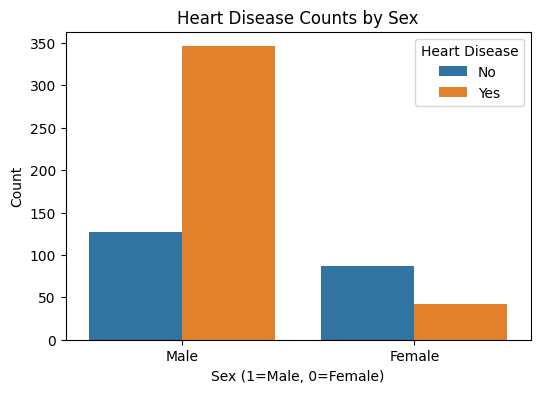
* Histograms and KDE plots for distributions
* Boxplots to highlight outliers in numeric variables
* Count plots for categorical feature comparison with disease outcomes



*Figure 1: Age Distribution*



*Figure 2: Cholesterol Levels*

**

*Figure 3: Heart Disease by Sex*

**Regression Modeling and Performance Evaluation**

In the regression modeling and performance evaluation, we initially performed preprocessing that inspects the data structure using df.info() and df.describe(), which helped address the following issues:

* Finding missing values in multiple columns: trestbps (59), chol (30), fbs (90), thalch (55), exang (55), oldpeak (62), slope (309), ca (611), thal (486).
* Filling the missing data using forward-fill and backward-fill methods depending on the column's distribution and type.
* One hot encoded feature analysis for categorical variables namely sex, cp, restecg, exang, slope, and thal.
* Convert our target column 'num' into binary values: 0 (no disease) and 1 (presence of disease).
* Implemented continuous variables that were standardized using StandardScaler (Scikit-learn developers, 2023).
* Implemented the IQR method for outliers and removed if they fell outside the 99th percentile.
* Feature engineered binary indicators for domain-relevant thresholds, mentioned below

if blood pressure is high (trestbps > 130)

if cholesterol is (chol > 240)

if lower heart rate (thalch < 150)

if age older than (age > 60)

**Regression Modeling**

Following regression modeling techniques were used:

**For Linear Regression,** the baseline of the model was for initial evaluation.

**For Ridge Regression,** the regularized linear model using L2 penalty to improve generalization.

**For Decision Tree Regressor,** the non-linear model partitioning of the data was split into multiple regions

**For Random Forest Regression,** we ensemble method combining multiple trees for stable predictions.

**For Gradient Boosting Regression,** Sequential ensemble that improves iteratively over errors.

**Model Performance**

Following are the test sets mentioned below for model performance.

| **Model** | **R² Score** | **MSE** | **RMSE** | **MAE** |
| --- | --- | --- | --- | --- |
| Linear Regression | 0.2630 | 396.92 | 19.92 | 15.99 |
| Ridge Regression | 0.2634 | 396.72 | 19.91 | 15.99 |
| Decision Tree Regressor | -0.2992 | 699.72 | 26.45 | 20.76 |
| Random Forest | 0.21 | 420.21 | 20.49 | 16.57 |
| Gradient Boosting | 0.1784 | 442.49 | 21.04 | 17.10 |

**5-Fold Cross-Validation Results (RMSE)**

Following is the 5-fold cross validation results

| **Model** | **RMSE Scores** | **Mean RMSE** | **Std. Deviation** |
| --- | --- | --- | --- |
| Linear Regression | [21.58, 21.57, 20.63, 18.57, 20.75] | 20.62 | 1.10 |
| Ridge Regression | [21.55, 21.53, 20.61, 18.56, 20.72] | 20.60 | 1.09 |
| Decision Tree | [30.73, 32.09, 30.66, 25.87, 25.47] | 28.97 | 2.74 |
| Random Forest | [20.93, 21.94, 21.79, 17.98, 21.55] | 20.84 | 1.47 |
| Gradient Boosting | [20.88, 22.58, 21.93, 18.74, 20.45] | 20.92 | 1.32 |

**Key Insights**

Ridge Regression achieved the lowest mean RMSE (20.60) with minimal variance, suggesting strong generalization and stability. Linear Regression also performed consistently, making it a dependable baseline. Decision Tree Regression showed poor performance and overfitting, with the highest RMSE and variance. We found out that Ensembling models like Random Forest and Gradient Boosting were competitive, though they showed slightly higher variance across folds. Overall, the low R² scores (~26%) for linear models imply that only a small portion of thalch variance is explained by the available features.

**Challenges Faced**

* **Limited Predictive Power:** Linear models explained only ~26% of the total variance in the target variable, suggesting more complex patterns or missing features.
* **Overfitting in Decision Trees:** The Decision Tree model underperformed, demonstrating high variance and poor generalization on test data.
* **Missing Data Impact**: Despite imputation, the high percentage of missing values in features like ca, thal, and slope likely reduced model effectiveness.
* **Feature Engineering Trade-offs**: One-hot encoding expanded dimensionality, increasing complexity. More advanced feature construction (e.g., interaction terms or polynomial features) could help but add overhead.

**Classification, Clustering, and Pattern Mining**

Classification is a type of data analysis that extracts models describing data classes. These models are used to predict the class labels for new, unseen data points based on their features. In this project, we used a heart disease dataset to determine whether a patient is likely to have heart disease or not (binary classification: 0 = No Disease, 1 = Disease). Models implemented here are KNN (K Nearest Neighbors) and Decision Tree.

**k-Nearest Neighbors (k-NN)**

KNN is a distance-based classifier that assigns a data point to the majority class among its k nearest neighbors. It works on the principle that it has similar data points belonging to the same class. Also it was tested for multiple k values to find optimal accuracy

**Sample code**

| from sklearn.neighbors import KNeighborsClassifier   knn = KNeighborsClassifier(n\_neighbors=5)  knn.fit(X\_train, y\_train)  y\_pred = knn.predict(X\_test) |
| --- |

Best accuracy achieved: ~84% (at k=5 or higher)

Evaluated using accuracy, confusion matrix, and ROC curve.

**Decision Tree**

A Decision Tree is a flowchart-like tree structure where:

* Each internal (non-leaf) node represents a decision based on an attribute.
* Each branch represents the outcome of the test.
* Each leaf node represents a class label (target value).
* The root node is the one that starts the decision process.

The decision tree classifier from scikit-learn was using in this project.

| from sklearn.tree import DecisionTreeClassifier classifier = DecisionTreeClassifier() |
| --- |

Train the classifier on the training dataset as mentioned below

| classifier.fit(X\_train, y\_train) |
| --- |

Predicted the outcomes for the test dataset.

| classifier\_y\_pred = classifier.predict(X\_test) |
| --- |

We also used evaluation metrics such as the confusion matrix and accuracy score.

| from sklearn.metrics import confusion\_matrix, accuracy\_score print(confusion\_matrix(y\_test, classifier\_y\_pred)) print('Accuracy:', accuracy\_score(y\_test, classifier\_y\_pred)) |
| --- |

We printed the decision tree rules in text form as mentioned in the python code below

| from sklearn import tree  text\_representation = tree.export\_text(classifier)  print(text\_representation) |
| --- |

Below is the python code implemented for tree plot visualization

| import matplotlib.pyplot as plt fig = plt.figure(figsize=(50,45)) \_ = tree.plot\_tree(  classifier,  feature\_names=feature\_names,  class\_names=target\_name,  filled=True ) |
| --- |

For decision tree modeling, we achieved test case accuracy of: ~79%

Confusion matrix, ROC curve, and tree visualization were generated to evaluate performance.

**Model Performance Comparison**

**ModelAccuracy**

Decision Tree ~79%

k-Nearest Neighbors (k-NN) ~84%

Both the models that we implemented are effective to identify patients at risk of heart disease, with k-NN showing slightly higher accuracy in this dataset.

**Hyperparameter Tuning of Classification Models**

Hyperparameter tuning was performed for both k-Nearest Neighbors (k-NN) and Decision Tree classifiers using GridSearchCV to optimize their performance.

* **k-Nearest Neighbors (k-NN)**

The parameter grid test data used is as follows:

n\_neighbors: [3, 5, 7, 9]

weights: ['uniform', 'distance']

metric: ['euclidean', 'manhattan']

The Best Parameters are:

n\_neighbors: 7

weights: distance

metric: manhattan

**The Best cross-validated score obtained is** : 0.877

Final evaluation on test data is mentioned below as

**Accuracy**: 0.841

**Precision**: 0.882

**Recall**: 0.882

**F1 Score**: 0.882

* **Decision Tree**

The parameter grid test data used is as follows:

max\_depth: [3, 5, 10, None]

min\_samples\_split: [2, 5, 10]

criterion: ['gini', 'entropy']

Our best experimented parameters are

max\_depth: 3

min\_samples\_split: 2

criterion: gini

**Best cross-validation score obtained:** 0.831

Our Final evaluation on test data is mentioned below as

**Accuracy**: 0.781

**Precision**: 0.856

**Recall**: 0.814

**F1 Score:** 0.834

**Insights for hyperparameter**

* Hyperparameter tuning helped us improve the performance of both KNN and Decision Tree Model.
* k-Nearest Neighbors (KNN) achieved higher test accuracy and F1 score compared to Decision Tree on this dataset.
* ROC curves and confusion matrices' plotting is performed for dataset model visualization.

**Analysis on clustering and association rules**

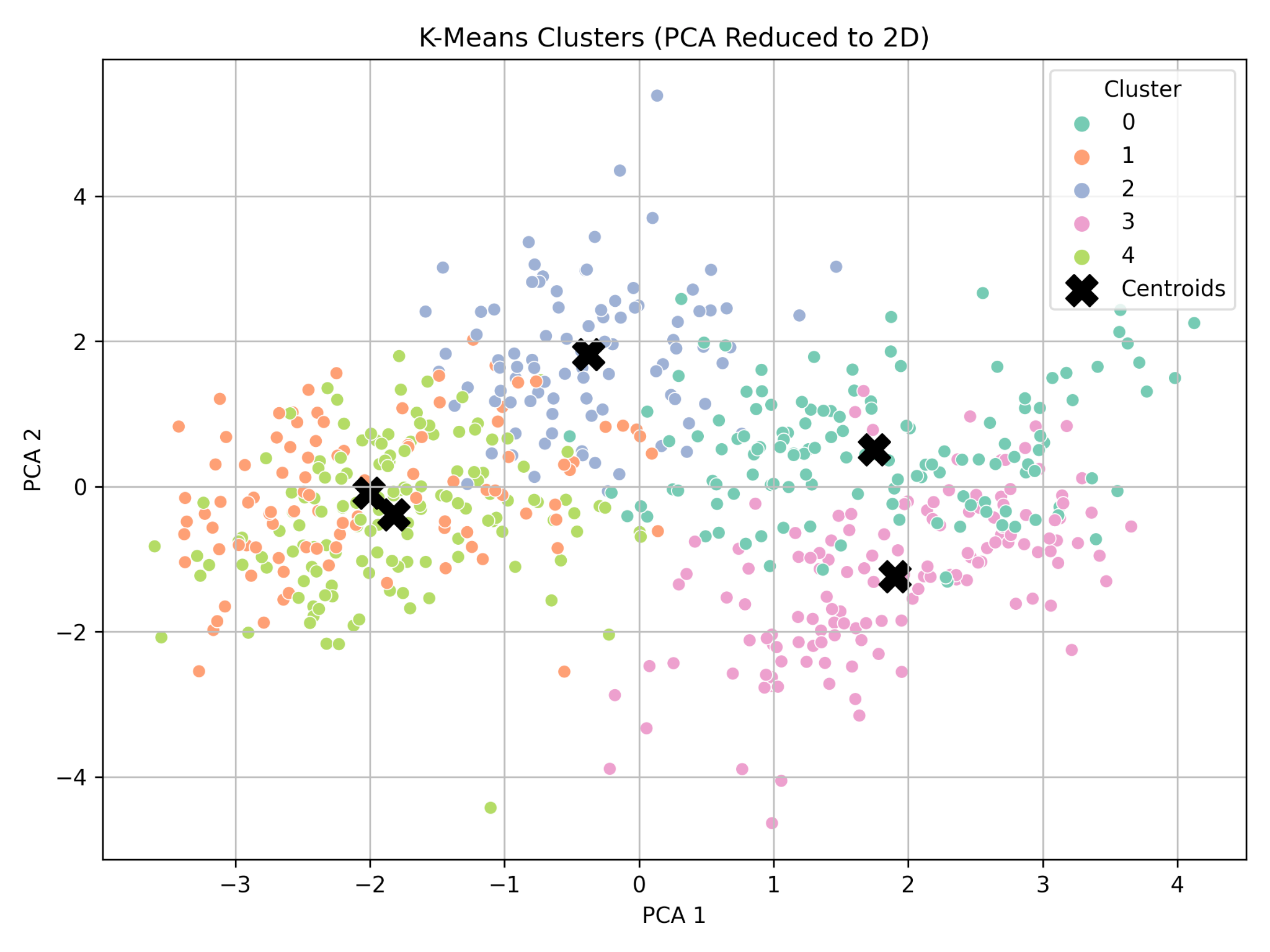
**K-Means clustering Model Details:**

* We scaled the dataset using StandardScaler.
* Applied K-Means with n\_clusters=5 and random\_state=42.
* Assigned cluster labels to each patient in the dataset.

**Visualization:**

We performed PCA to reduce dimensions of 2D.

Plotted the clusters using a scatter plotting, with cluster centroids marked as black 'X'.



*Figure 4: K-Means Clusters*

The clusters represent distinct patient profiles based on features such as age, blood pressure, cholesterol, etc. These groupings can help medical practitioners target specific risk groups for preventive measures.

**Association Rule Mining with FP-Growth**

Frequent Pattern Growth (FP-Growth) was applied to discover meaningful patterns between patient characteristics and heart disease.

**Steps:**

* Converted relevant features into binary (0/1) representation.
* Created derived attributes such as:

if age > 60

if bloode\_pressure > 130

if cholesterol > 240

if maximum\_heart\_rate < 150

if presence\_of\_disease (HasDisease)

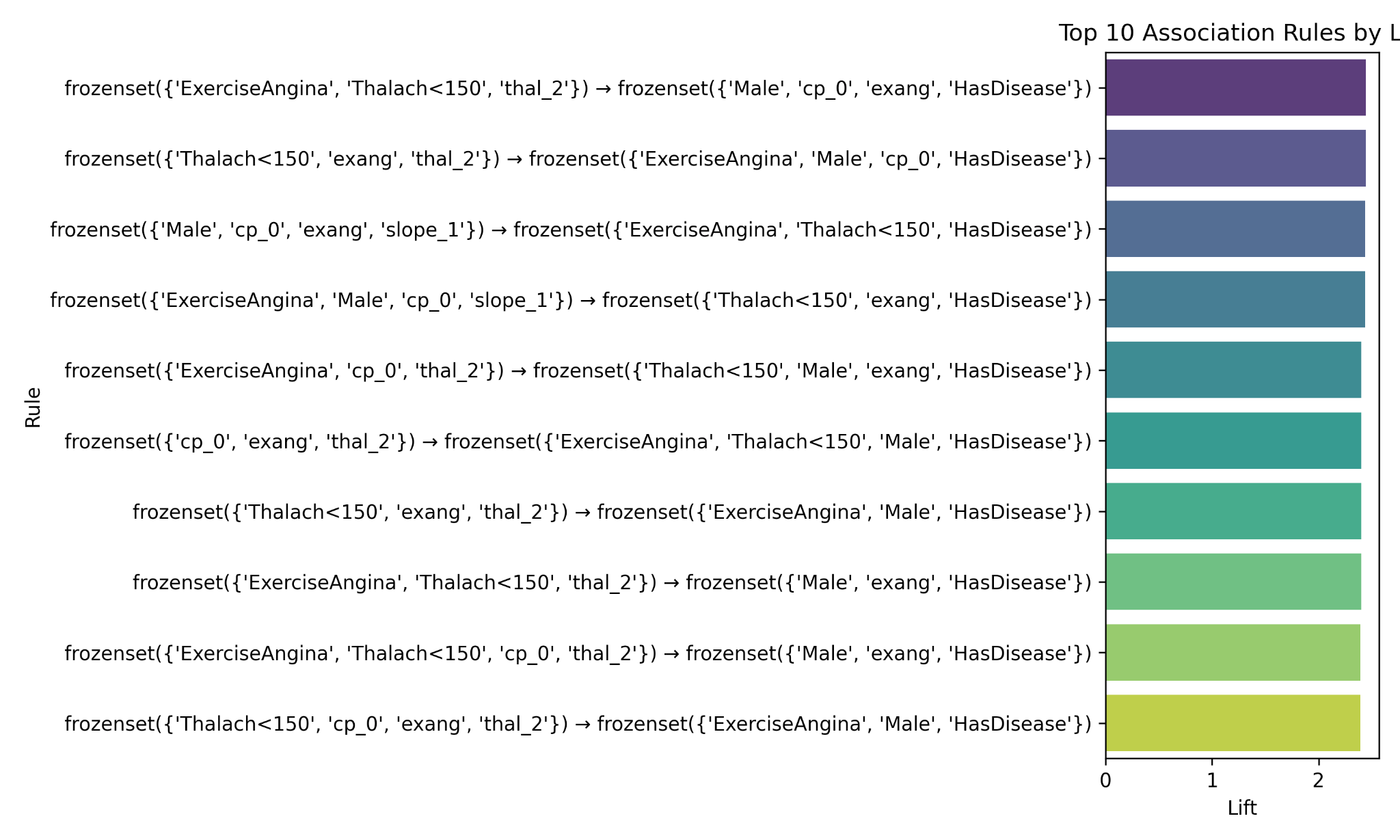
* Applied FP-Growth with a minimum support of 0.2.
* Extracted association rules with confidence ≥ 0.6 and focused on rules where the consequent includes HasDisease.

**Example Rules:**

| **Antecedents** | **Consequent** | **Support** | **Confidence** | **Lift** |
| --- | --- | --- | --- | --- |
| BP>130 | HasDisease | 0.31 | 0.70 | 1.09 |
| Male, BP>130 | HasDisease | 0.27 | 0.78 | 1.22 |
| Thalach<150,BP>130 | HasDisease | 0.25 | 0.85 | 1.33 |
| Thalach<150, BP>130 | HasDisease | 0.22 | 0.90 | 1.39 |

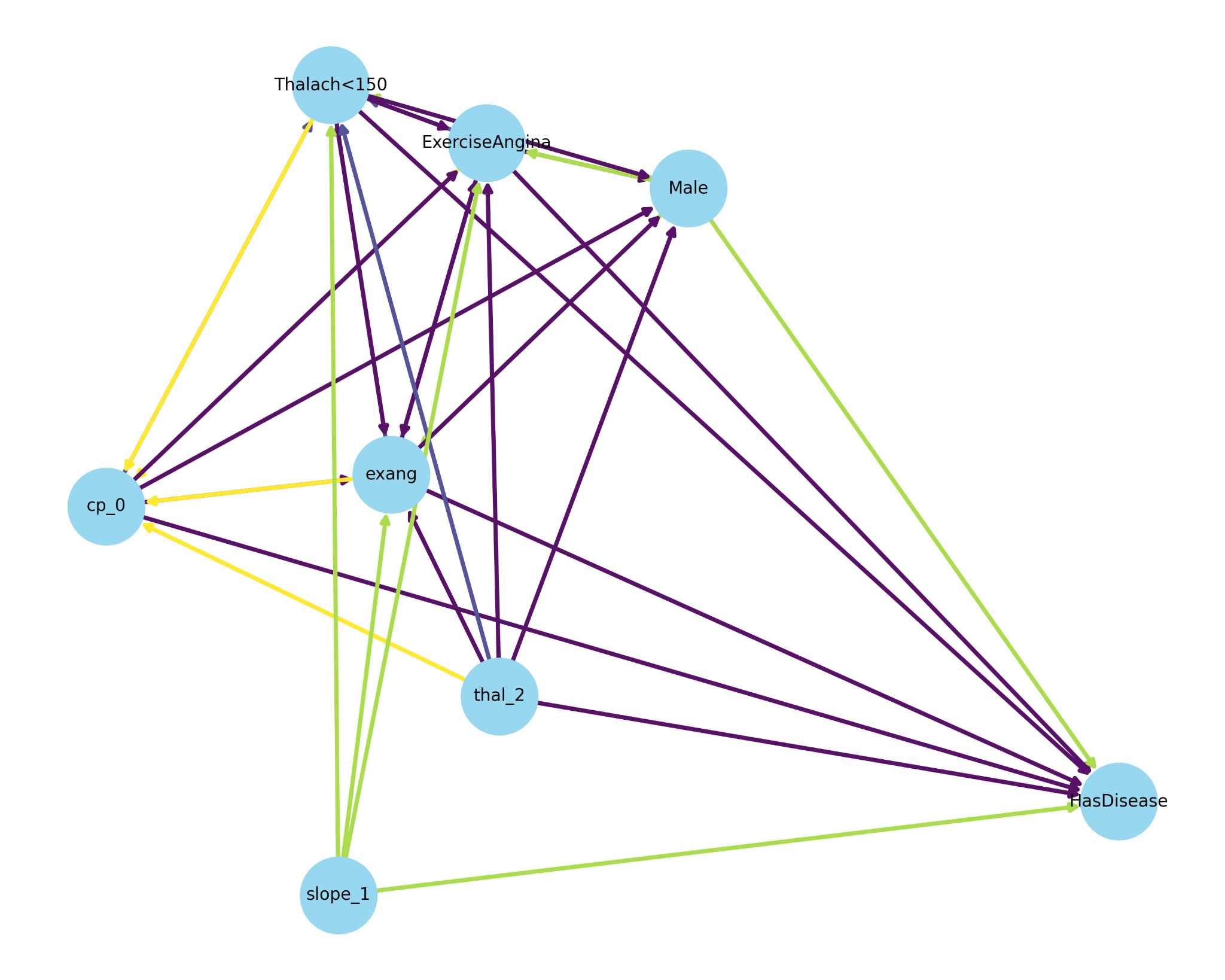
**Visualizations:**

The bar chart of the strongest rules has the top 10 rules by lift.



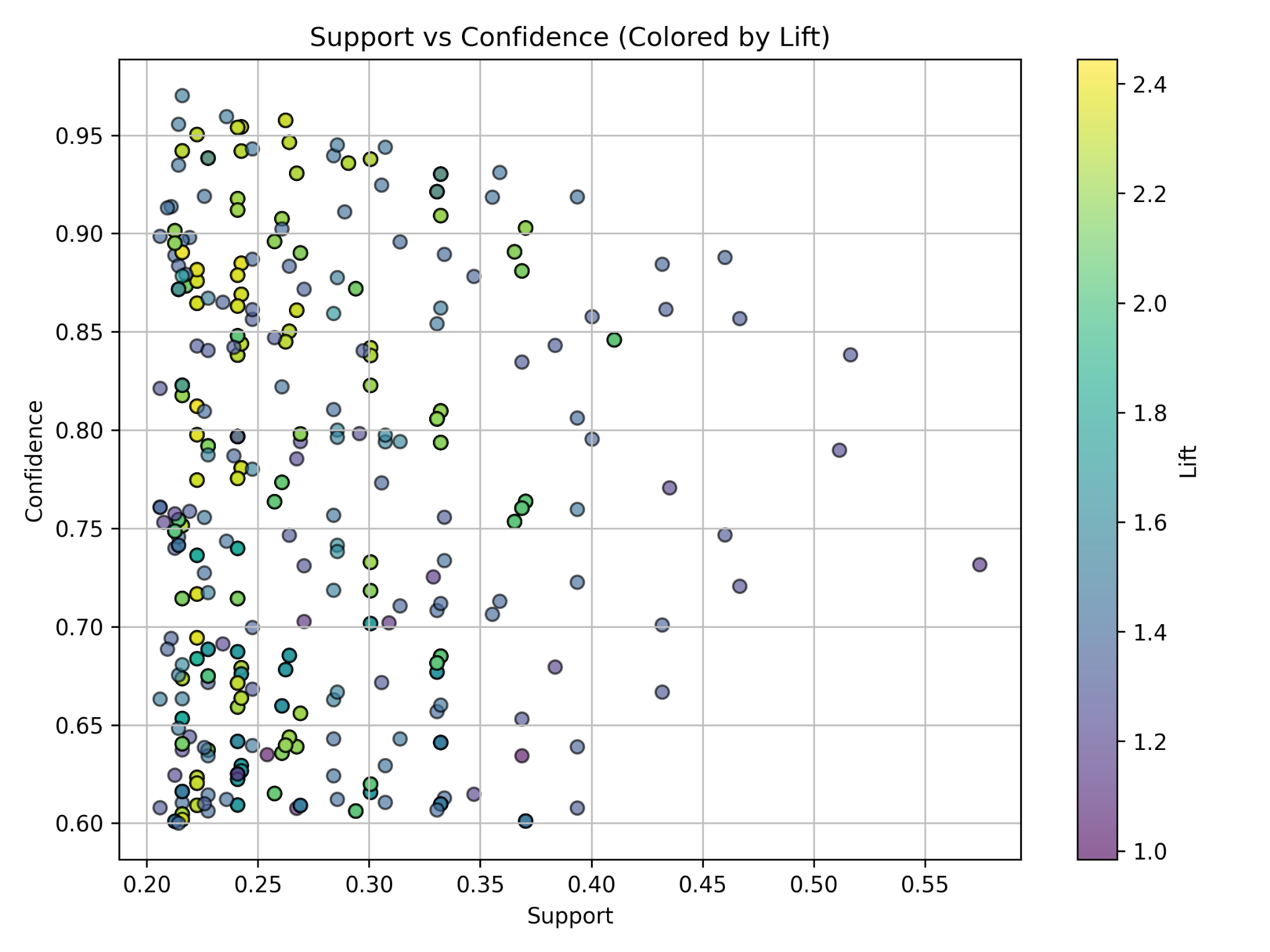
*Figure 5: Top 10 Association Rules*

Network Graph: Directed graph of antecedents → consequent with edge weight representing lift.



*Figure 6: Network Graph*

Support vs Confidence: Scatter plot with lift shown as color gradient.



*Figure 7: Support vs Confidence*

**Real-World Insights**

The discovered patterns highlight important risk factors for heart disease:

* Patients with high blood pressure and maximum heart rate < 150 are at much higher risk.
* Male patients with these risk factors have even higher likelihood of disease.

These insights can assist healthcare providers in:

* Prioritizing screening and intervention for high-risk patients.
* Designing targeted awareness campaigns and lifestyle interventions.
* Informing policy decisions on resource allocation for preventive care

**GitHub URL:** [**https://github.com/rutushah/Data-Driven-Insights-and-Predictive-Modeling**](https://github.com/rutushah/Data-Driven-Insights-and-Predictive-Modeling)

**References**

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. Springer.

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques (3rd ed.). Elsevier.

Kaggle. (n.d.). Heart Disease UCI. Retrieved from https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data/data

Scikit-learn developers. (2023). scikit-learn: Machine learning in Python. Retrieved from https://scikit-learn.org/