Using SEMMA to Diagnose Coronary Heart Disease

Steven Chang

College of Engineering, San Jose State University

Abstract

SEMMA is a methodology for performing data mining. It involves five phases: sample, explore, modify, model, and assess. In this paper, we will explore the SEMMA methodology by performing principled, step-by-step, phase-by-phase data science with the process. To do so, we will demonstrate its usage in a project. In this project, we will take the role of a data scientist working for a medical company. Our goal is to be able to determine if a patient is likely to have coronary artery disease so that we can order more thorough tests to properly diagnose the patient. Medical tests can be invasive and costly so we want to minimize the number of tests ordered, however, we do not want to let a patient go undiagnosed. We will be using the SEMMA methodology for data mining to create a model capable of classifying patients with or without Coronary Artery Disease.

Phase 1: Sample

In this step, we select a data set that can help us achieve this goal. We selected our dataset from Kaggle. The data set we found included information about a patient's demographics, symptoms as well as medical test results. Because the purpose of the model we wish to train is to identify if a test is required, I dropped the columns of data that represented test results as we would not have access to that data when determining if a test is advisable.

This dataset consists of 303 rows and 32 columns. Given the dataset's size, it is manageable, and we don't need to sample a smaller subset.

Dataset Overview:

The dataset has columns such as Age, Weight, Length, Sex, BMI, and several others that represent the patient's demographics, symptoms, and risk factors.

The target variable is the Cath column, which indicates whether a patient has coronary artery disease ("Cad") or is normal.

Phase 2: Explore

In this phase, we will delve deeper into the dataset to better understand its characteristics and distributions. To do so we will get a summary of the dataset's statistics, check for missing values, and visualize distributions of key variables.

Dataset Statistics Summary

Age: The patients' ages range from 30 to 86 years, with an average age of approximately 58.9 years.
Weight: The weights of the patients range from 48 to 120 kg, with an average weight of roughly 73.8 kg.
Length (Height): The heights of the patients range from 140 to 188 cm, with an average height of about 164.7 cm.
BMI: The Body Mass Index (BMI) values range from 18.1 to 40.9, with an average BMI of around 27.2.
DM (Diabetes Mellitus): This is a binary variable (0 or 1) indicating the presence of diabetes. About 29.7% of the patients have diabetes.
HTN (Hypertension): This is also a binary variable. About 59.1% of the patients have hypertension.
Current Smoker: 20.8% of the patients are current smokers.
EX-Smoker: 3.3% of the patients are ex-smokers.
FH (Family History): 15.8% of the patients have a family history of coronary artery disease.
BP (Blood Pressure): The average blood pressure is approximately 129.5, with values ranging from 90 to 190.
PR (Pulse Rate): The pulse rates range from 50 to 110, with an average of approximately 75.1.

Edema: This binary variable indicates the presence of edema. Only about 3.9% of the patients have edema.
Typical Chest Pain: 54.1% of the patients experience typical chest pain.
Function Class: This seems to be an ordinal variable with values ranging from 0 to 3. Its meaning will need to be further clarified.

There are no missing values in any of the columns. This means we don't have to deal with imputation or other methods to address missing data at this stage.

Data Visualizations

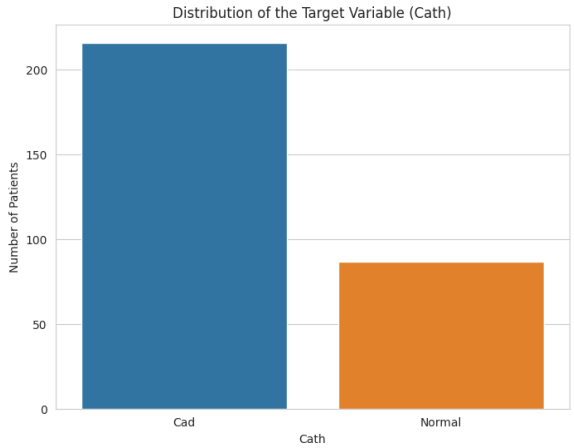

Figure 1. Distribution of Target Variable

From the plot, we can observe the distribution of the target variable, "Cath". There are two categories: Cad (indicating coronary artery disease) and Normal. The dataset seems fairly imbalanced between the two categories, meaning that re-balancing may need to take place.

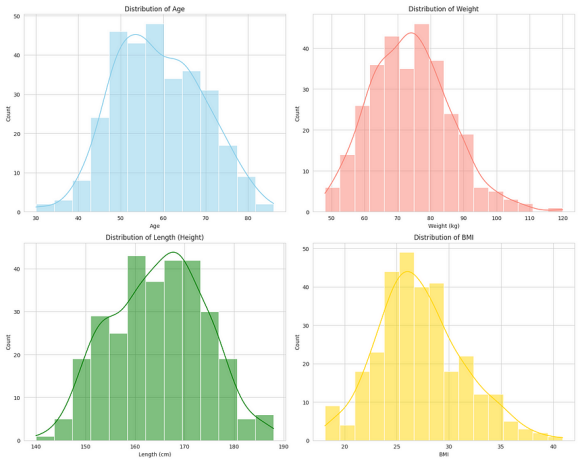

Figure 2. Distribution of Numerical Variables

Based on the histograms above, we can make the following observations on the dataset.
Distribution of Age: Most of the patients are aged between 50 and 70 years.
Distribution of Weight: The weights of the patients seem to be fairly normally distributed, with a slight right skew. Most patients weigh between 60 and 90 kg.
Distribution of Length (Height): The majority of patients have a height between 155 cm and 175 cm.
Distribution of BMI: The Body Mass Index (BMI) distribution is slightly right-skewed, with most patients having a BMI between 23 and 30.

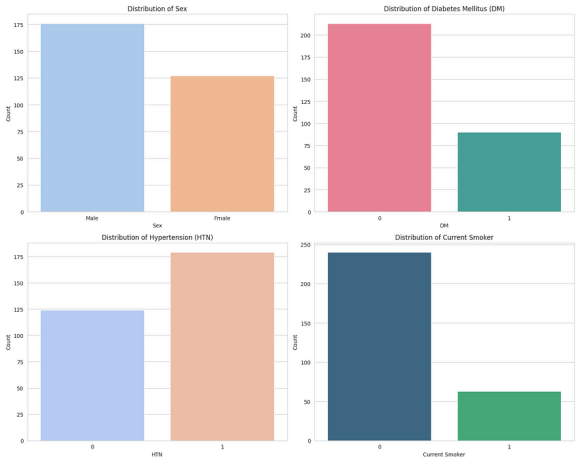

Figure 3. Distribution of Categorical Variables

Distribution of Sex: There are more male patients than female patients in the dataset.
Distribution of Diabetes Mellitus (DM): A majority of the patients do not have diabetes, as indicated by

the value “0”. However, a considerable portion does have diabetes (value “1”).
Distribution of Hypertension (HTN): More patients in the dataset have hypertension (value “1”) than those who don’t.
Distribution of Current Smokers: A majority of the patients are not current smokers, but there’s still a sizable portion that smokes.

Phase 3: Modify

In the Modify phase, we’ll transform the data to better suit the modeling process. This includes encoding categorical variables, scaling numerical features, and handling potential outliers or anomalies. This will involve encoding categorical variables to numerical format, scaling continuous features to bring them to a similar scale, and handling potential outliers or anomalies if identified.

Encoding Categorical Values

Many models require data in a numerical format. Categorical data, including multi-categorical data points, must be converted into a numerical format to be used. We utilized one-hot encoding to convert categorical variables into a numerical format that is acceptable for training a model.

Scale Numerical Features

Scale numerical variables so they have similar scales, which can help with certain machine learning algorithms. The numerical columns we’ll scale are Air temperature [K], Process temperature [K], Rotational speed [rpm], Torque [Nm], and Tool wear [min]. We’ll also scale the newly derived feature Total_Specific_Failures.

Check for Outliers

Weight, Blood Pressure, BMI and Pulse Rate all have a handful of outliers. However, given the medical context of the data, it’s crucial to be cautious about removing outliers, as they could represent genuine cases. Instead of removing them, we might consider other methods like robust scaling or using models that are less sensitive to outliers.

Phase 4: Model

In the Model phase, we’ll select an appropriate classification algorithm, train an initial model, and evaluate its performance. To do this we will split the data into training and verification steps, choose a classification algorithm, train the model, and evaluate its performance.

To train the model we split the dataset into a training set of 242 samples and a verification set of 61 samples. The split was done in an 80–20 ratio, ensuring that the validation set is representative of the overall data distribution.

Initial Model — Random Forest Model:

We initially created a Random Forest Model An ensemble learning method that can capture complex patterns. Random Forest is an ensemble learning method that combines multiple decision trees to produce a more accurate and robust prediction. Given its ability to capture complex relationships and its inherent feature importance estimation, it’s a popular choice for many classification tasks.

Using the Random Forest classifier, we achieved the following results on the validation set:

Accuracy: Approximately 81.97%

Confusion Matrix:

True Negatives (TN): 40
False Positives (FP): 3
False Negatives (FN): 8
True Positives (TP): 10

Classification Report:

Precision (Class 0): 0.83
Recall (Class 0): 0.93
F1-score (Class 0): 0.88
Precision (Class 1): 0.77
Recall (Class 1): 0.56
F1-score (Class 1): 0.65

Given our initial requirement where false positives are preferable to false negatives (we would rather order an unnecessary test than miss a patient with the disease), it’s important to focus on the False Negatives (FN) in the confusion matrix. Currently, there are 8 FN, meaning 8 patients with the disease were missed by the model. We will test other models to see if they perform better.

Logistic Regression Model

Accuracy: Approximately 83.61%

Confusion Matrix:

True Negatives (TN): 41
False Positives (FP): 2
False Negatives (FN): 8
True Positives (TP): 10

Classification Report:

Precision, Recall, and F1-score for both classes are relatively balanced, with the model having slight difficulty in detecting true positives (patients with the disease).

The Logistic Regression model provides a balanced performance with fewer false negatives compared to the default Random Forest model.

Gradient Boosted Machines

Accuracy: Approximately 81.97%
Confusion Matrix:

True Negatives (TN): 39
False Positives (FP): 4
False Negatives (FN): 7
True Positives (TP): 11

Classification Report:

The model provides balanced precision, recall, and F1-score values for both classes. The recall for positive cases (patients with the disease) is around 61%, indicating the model's capability to detect these cases.

Both Logistic Regression and GBM models have similar accuracy scores.The GBM model has one fewer false negative compared to the Logistic Regression model, which is a slight advantage given our objective of minimizing false negatives.

Phase 5: Assess

In the assessment phase, we evaluate the model's performance more deeply and consider its implications from a business perspective.

Comparison of Models

Random Forest: Accuracy: ~81.97%, False Negatives (FN): 8
Logistic Regression: Accuracy: ~83.61%, False Negatives (FN): 8
Gradient Boosting Machines (GBM): Accuracy: ~81.97%, False Negatives (FN): 7

Business Implications

False Positives: These represent patients who are incorrectly classified as having coronary artery disease. From a medical perspective, this might lead to unnecessary tests, which can be costly and cause anxiety for the patient. However, given the serious nature of the disease, it's preferable to err on the side of caution.

False Negatives: These are more concerning as they represent patients who have the disease but are not identified by the model. Missing such cases can have severe medical consequences.

Deployment Considerations

Due to the relatively high rates of false negatives, if deployed in a clinical setting, the model should be used as a supplementary tool and not replace medical judgment.

Regularly retrain the model with new data to ensure it remains accurate and relevant.

Monitor the model's performance in the real world and gather feedback from clinicians to understand its strengths and weaknesses.

Conclusion

Given the data and models we've explored, the GBM model offers the lowest number of false negatives. Still, it's essential to consider both the statistical performance and the clinical context when making decisions. The model can be a valuable tool for screening patients, but medical professionals should have the final say in diagnostic decisions.