**Using Voting Classifier to identify Heart Attacks**

# Dataset: Heart Attack Analysis & Prediction Dataset

# David Jones, [jjones12@bellarmine.edu](mailto:jjones12@bellarmine.edu)

# *Abstract*

# This project uses Voting Classifier to identify the probability of a heart attack occurring using several classification techniques. This project uses Voting Classifier to compare four different classification techniques to determine which model fits the dataset best. This project uses the libraries pandas, NumPy, seaborn, and matplotlib.

# Introduction

# The classification techniques used in this project are Logistic Regression, SVC, Decision Tree, and Random Forest. The dataset uses the variables: Age, Sex, exang, ca, cp, trtbps, chol, fbs, rest\_ecg, and thalach to determine the chances an individual has a higher or lower chance of a heart attack.

# I chose this dataset because the data worked well with the goal I was trying to achieve. The dataset had no missing values, so the data was easy to work with. I chose the Voting Classifier method to determine which classifier model gave the best score.

# Background

# Data Set Description

# The dataset can be found on Kaggle’s website [1]. The goal of the dataset is to use variables to determine if an individual has a higher or lower chance of a heart attack. Medical professionals use Heart Attack Analysis datasets to determine the probability an individual will have a heart attack. The data in this model was collected by Rashik Rahman [1]. The data was collected so people could practice machine-learning classification methods.

# Machine Learning Model

# This project compared four different machine learning classification methods to determine which fit the dataset obtained from Kaggle the best. The four classification methods tested were Logistic Regression, Support Vector Classification (SVC), Decision Tree Classification, and Random Forest Classification. These four classification methods are all contained within the sklearn library.

# Logistic Regression analyses data and outputs a binary value (0 or 1). It does this by analyzing the relationship between independent variables. Examples of a problem that Logistic Regression can solve well are whether an email is spam or if a student will get into their school of choice. The model will output a score between 0% and 100%. If the predicted value is above 50% it will be labeled as “1” and “0” if the value is below 50%. Support Vector Classification is well suited for small/medium complex datasets. It takes a ‘gamma’ parameter which has a default value of ‘auto’. SVC works by splitting the class into two classes and creating a plane in which these two classes overlap.

# Decision Tree Classification (DTC) uses decision trees to create a model that predicts an outcome based on data features. DTC is easy to use as it does not require much data preparation such as normalization or dummy variables. Random Forest Classification is part of Sklearn’s ensemble library. The model works by splitting the model into several decision trees and compares their outputs by voting on them.

# Exploratory Analysis

# The dataset I chose had 303 samples with 14 columns of various data types. The table below shows the variables in my dataset and their respective types. The data correlated well as the variables in the dataset were all important to produce the output.

# Table 1: Variables

|  |  |
| --- | --- |
| Variable Name | Data Type |
| age | int64 |
| sex | int64 |
| cp | int64 |
| trtbps | int64 |
| chol | int64 |
| fbs | int64 |
| restecg | int64 |
| thalachh | int64 |
| exng | int64 |
| oldpeak | float64 |
| slp | int64 |
| caa | int64 |
| thall | int64 |
| output | int64 |

# Methods

# Dataset preparation consisted of creating a train/test split and scaling the dataset to work with the voting classifier model. The dataset did not have any missing values and did not need any dummy variables created as seen in the table above. All variables were numerical data types.

# Data experiments for this project consisted of changing the parameters for the classification models to change their output score. For Logistic Regression, the only parameter that could be changed was max\_iter. Random Forest Classifier did not have any parameters to manipulate so I was unable to change its output. SVC had one parameter that could be manipulated being gamma which was set to auto by default. Decision Tree Classifier had 2 parameters that could be manipulated, criterion and max\_depth.

# The following tools were used for this analysis: Python running on Google Colaboratory was used for all the analysis and implementation. Scikit-learn 1.1.2, Pandas 0.22.0, matplotlib 3.2.2, and Seaborn 0.11.2. pandas were used to read the .csv file for the dataset, Scikit-learn was used to implement the various classification models used for the project, matplotlib was used to visualize the data in the form of graphs, and seaborn was for data visualization.

# Results

# The results of this project concluded that SVC worked the best for the dataset chosen for this project. The best model for this SVC had a score of 87%. The worst model for this dataset was Decision Tree Classifier with a score of 82%. SVC worked the best for this model as SVC is the best model for classification problems for the scale of my dataset. Decision Tree Classifier was the worst model for my dataset because the dataset is highly correlated and did not work well with decision trees.

# One of the main problems I encountered was finding a suitable dataset for the four classification models used. I also had trouble determining which variables were important to the data and which could be dropped. In order to improve the accuracy of my data I would need a dataset with more data. The dataset I chose had 303 samples per column so there was not much data for the model to analyze.

# Conclusion

# In conclusion, this project found that Support Vector Classification is the best model for the dataset chosen. The model was good, but it could be better with more tweaking of parameters to produce the best possible score. The dataset was not overfitted so the scores were accurate based on the data that was used.

# References

[1]: Rahman, Rashik. “Heart Attack Analysis & Prediction Dataset.” *Kaggle*, 22 Mar. 2021, <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>.