# Heart Failure & Stroke Database & API

## ETL Project - Group 7

Team Members- Gabe, Noah, Bailey, Renuka

## Overview

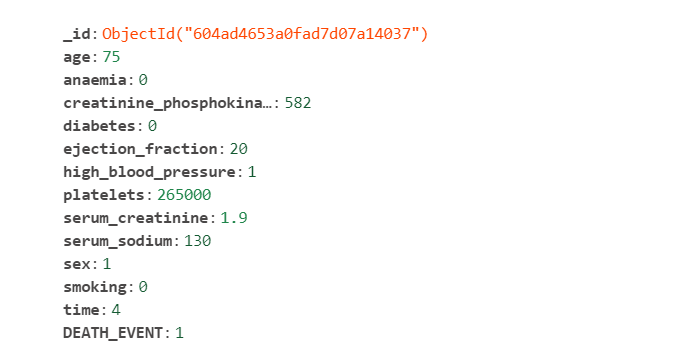
[According to the Centers for Disease Control](https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm) and Preventions’ National Center for Health Statistics, heart disease and stroke are the first and fifth most common cause of death in the United States. The aim of this project is to help inform the public of common heart and brain health metrics to be aware of, by generating a user-friendly API that can be queried by both medical professionals and laymen alike. Due to the disparate nature of the data collected between each dataset, a Non-Relational (NoSQL) database has been selected to store the data for ease of use and portability.

## Data Sources:

[Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/fedesoriano/stroke-prediction-dataset) - Datasets containing 11 clinical features for predicting stroke events.

[Heart Failure Dataset](https://www.kaggle.com/andrewmvd/heart-failure-clinical-data) - Data set for Heart Failure

## Data Attributes:

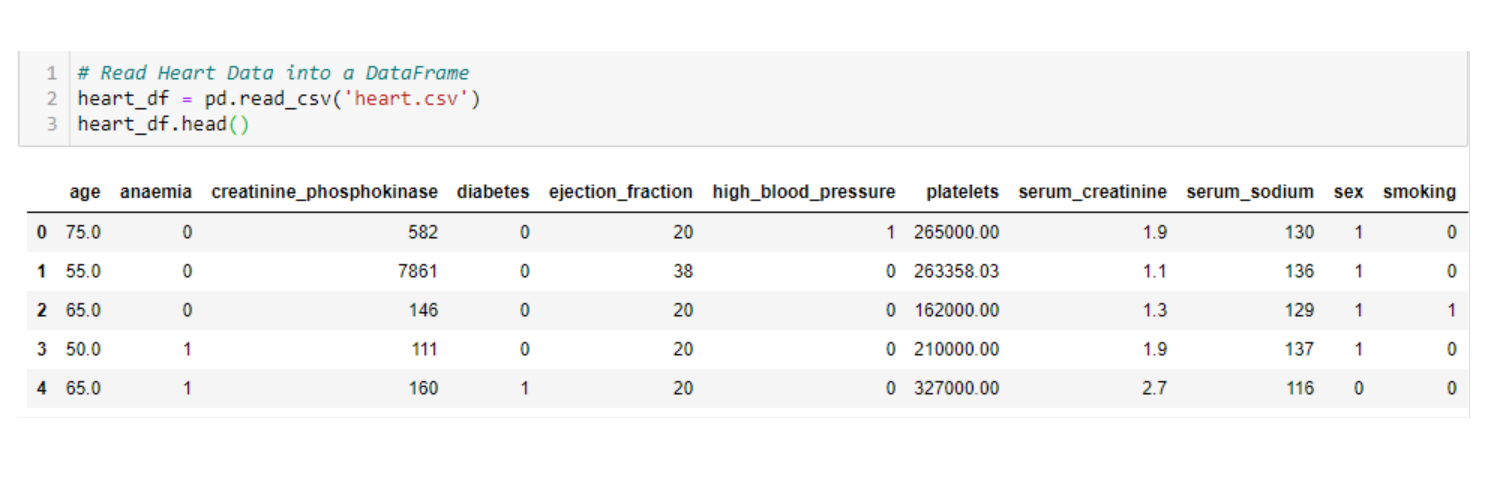


**Figure 1: Screenshot from MongoDB Heart\_Failure\_Record Collection**

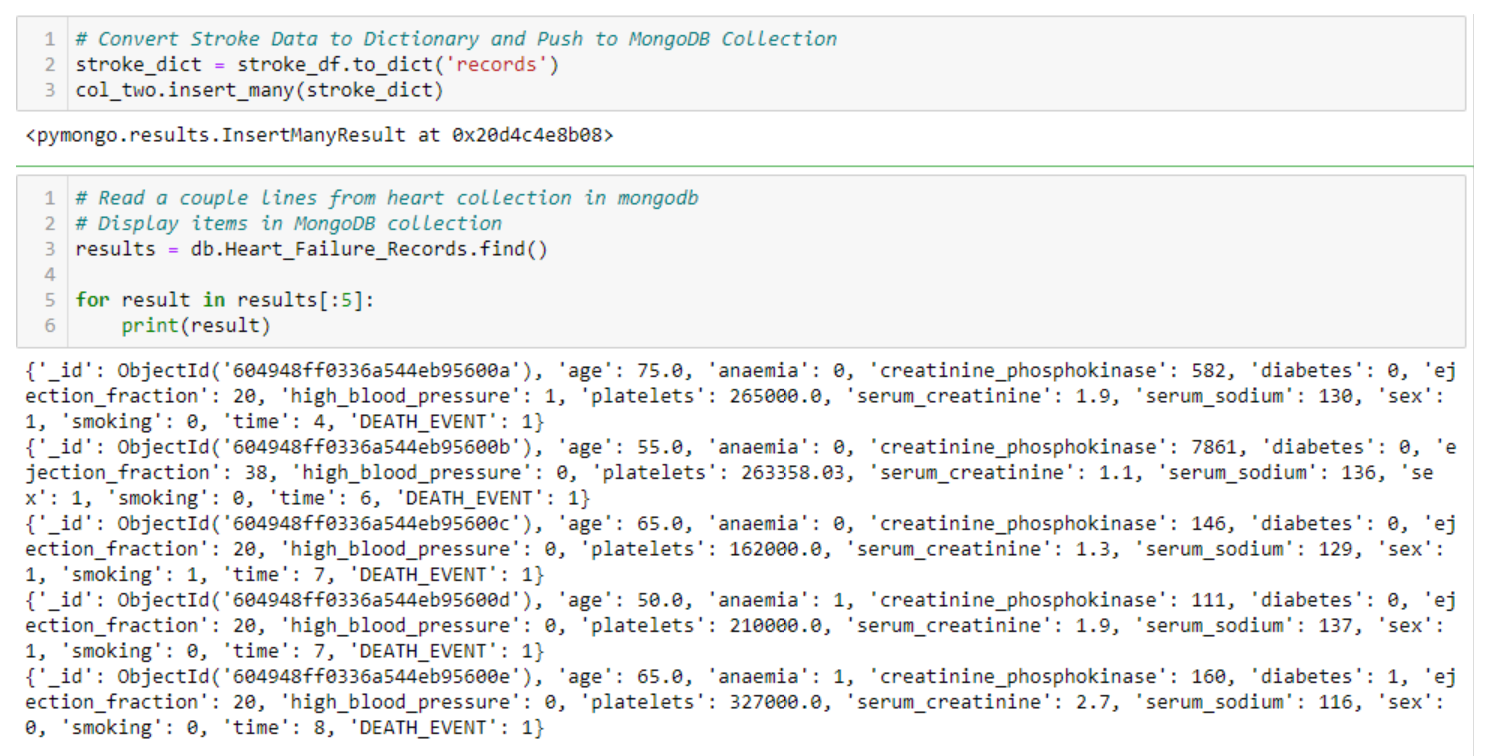


**Figure 2: Screenshot from MongoDB Stroke Collection**

## ETL Process Summary

Extract: Each dataset was downloaded from Kaggle.com in CSV format and subsequently loaded into a Jupyter Notebook for data munging and cleaning as required.

Transform: Data has been grouped by key categorical variables to produce common aggregated queries across each dataset.

Load: Using the pymongo module available for Python, insert raw and transformed tables into MongoDB to serve as a backend for a Flask API.

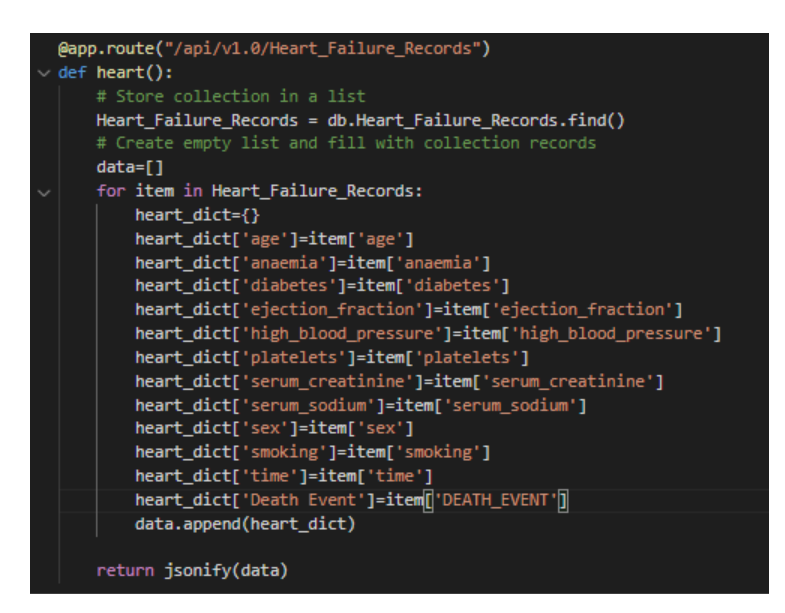
# API Documentation

An **application programming interface** (**API**) is a [computing interface](https://en.wikipedia.org/wiki/Interface_(computing)) that defines interactions between multiple software or mixed [hardware](https://en.wikipedia.org/wiki/Computer_hardware)-software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. It can also provide extension mechanisms so that users can extend existing functionality in various ways and to varying degrees.[[1]](https://en.wikipedia.org/wiki/API#cite_note-Fisher1-1)

The Heart Failure and Stroke Database API contains six easily accessible routes that can handle requests without the need for an access key or credentials, it is fully open sourced. The following is a list of the available API routes along with utilization parameters and output examples.

* Below route returns all heart records from database:
  + /api/v1.0/Heart\_Failure\_Records
* Below route returns all stroke records from database:
  + /api/v1.0/Stroke\_Records
* Below route returns heart failure death by mean age | Death = 1, Non-Death = 0:
  + /api/v1.0/Heart\_Failure\_by\_Age
* Below route returns strokes by mean age | Stroke = 1, no stroke = 0:
  + /api/v1.0/Stroke\_by\_Age
* Below route returns heart failure data by smoking or not | **Input 1 for Smoker, 0 for Non-Smoker**:
  + /api/v1.0/Heart\_Failure\_by\_Smoking/
* Below route returns stroke data by gender | **Input Male or Female**:
  + /api/v1.0/Stroke\_by\_Gender/

#### Example Flask API Route



### Example API Result Screen