README:

Contributors: Anna Puk and Jaya Deonandan

This project aims to examine epidemiological data, specifically the potential causes and risk factors that contribute to developing coronary heart disease (CHD) throughout a ten-year span.

**Input Data:**

The original data is available in .csv format. [Click here to access the Framingham Heart Study Dataset](https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset)

**Number of Instances:** 4240

**Number of Attributes:** 16 including the TenYearCHD target function

**Class Distribution:** 0 (did not develop CHD): 3596 (84.8%), 1 (developed CHD): 644 (15.2%)

Features of this data set include:

§ Male (Int) 0 = Female; 1 = Male

§ Age (Int) Age at exam time

§ Education (String) 1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college

§ currentSmoker (Int) 0 = nonsmoker; 1 = smoker

§ cigsPerDay (String) number of cigarettes smoked per day (estimated average)

§ BPMeds (String) 0 = Not on Blood Pressure medications; 1 = Is on Blood Pressure medications

§ prevalentStroke (Int)

§ prevalentHyp (Int)

§ diabetes (Int) 0 = No; 1 = Yes

§ totChol (String) mg/dL

§ sysBP (Int) mmHg

§ diaBP (Int) mmHg

§ BMI (String) Body Mass Index calculated as: Weight (kg) / Height(meter-squared)

§ Heartrate (String) Beats/Min (Ventricular)

§ Glucose (String) mg/dL

§ TenYearCHD (Int) 0 = No; 1 = Yes; identifies those that did or did not develop CHD during the study period

This notebook contains a comprehensive analysis of the dataset. It also contains functions to load, clean, and save the scrubbed data as inputs for the machine learning sections. Using Pipeline, GridSearchCV, and Cross-Validation with five folds we built five models:

o K-Nearest Neighbor (KNN)

o Logistic Regression

o Decision Tree Classification (using AdaBoost)

o Random Forest (RFC)

o Hard Voting

o Stacking

Each Machine Learning Model Process:

1. Both the unbalanced and balanced data sets have been split into train/test subsets
2. Fit training data
3. Predict with training data and run ‘Classification Report Summary’
4. Tune parameters of and then make predictions for both balanced and unbalanced data to get the best models for each for comparison

### Comparison of Models and Results:

In comparing our models, we chose to look at the ROC curves along with f1 and accuracy scores to compare and determine which models provided the best results.

### Conclusion:

The results indicate that the best models capable of making the most accurate predictions are Decision Tree Classification with AdaBoosting, Random Forest and using Stacking with f1 scores 0.93, 0.93 and 0.96, respectively for the positive CHD class, and accuracy scores 93%, 93% and 97%, respectively for the positive CHD classes.