Analysis Appendix

Data Preprocessing

The Heart Attack Risk dataset contains 24 features, which are categorized by two ways as below:

1. **Based on data type**
    - ***9 Binary categorical variables:*** Sex, Diabetes, Family.History, Smoking, Obesity, Alcohol.Consumption, Previous.Heart.Problems, Medication.Use, Hemisphere
        - Encoded variables: Sex, Hemisphere
        - Others were kept in numeric from when building a correlation matrix before being converted to factors for training models
    - ***3 Multi-category Variables:*** Diet, Country, Continent
        - Converted to numeric first when building a correlation matrix
        - Converted to factors and finally dummies for training models
    - ***10 Continuous numeric variables:*** Age, Cholesterol, Blood.Pressure, Heart.Rate, Exercise.Hours.Per.Week, Sedentary.Hours.Per.Day, Income, BMI, Triglycerides, Sleep.Hours.Per.Day
        - Feature engineering: Systolic and Diastolic converted from Blood.Pressure
    - ***2 Discrete Numeric Variables*:** Stress.Level, Physical.Activity.Days.Per.Week

2. **Based on functions**
    - ***6 Demographic Factors:*** Sex, Age, Country, Continent, Hemisphere, Income
    - ***6 Physiological & Clinical Variables:*** Cholesterol, Blood.Pressure (Systolic, Diastolic), Triglycerides, Heart.Rate, BMI, Obesity
    - ***5 Medical History:*** Diabetes, Family.History, Previous.Heart.Problems, Medication.Use, Stress.Level
    - ***7 Lifestyle Factors:*** Smoking, Alcohol.Consumption, Diet, Exercise.Hours.Per.Week, Physical.Activity.Days.Per.Week, Sedentary.Hours.Per.Day, Sleep.Hours.Per.Day

Grouping by **data type e**nsures proper encoding and scaling for modeling, while grouping by **function** allows for targeted analysis of how different factors (e.g., demographic, physiological, lifestyle) influence heart attack risk.

Additionally, we exported 2 final clean datasets with the same train and test split but different in how the 3 multi-category variables were handled, **data** with factors and **data1** with dummies as summarized in the table below:

data	8763 obs. of 26 variables
data.test	4382 obs. of 26 variables
data.train	3138 obs. of 26 variables
data1	8763 obs. of 52 variables
data1.test	4382 obs. of 52 variables
data1.train	3138 obs. of 52 variables

- We used **data1** (dummy-encoded) for models like k-NN, and SVM, as these algorithms require numerical input.
- We used **data** (factor-based) for tree-based models like Logistic Regression, Naive Bayes, Decision Trees, and Random Forests, as they can natively handle categorical variables without the need for dummy encoding, as well as for better visuals of the trees.

Subsetting Data

- Subseted the dataset for Italy, Japan, US, and China
- Filtered the training and testing dataset (data1.train/data1.test) to include only rows for specific countries
- Removed columns in training and testing dataset that have all 0 values. For example, the columns that have dummy values for different countries

Logistic Regression

Steps to perform the logistic regression model on original dataset:

1. Fit a Full Logistic Regression Model: a logistic regression model was trained using all available predictors in data.train.

2. Performed Stepwise Feature Selection (StepAIC): stepwise selection was applied using the Akaike Information Criterion (AIC). This method adds or removes variables to optimize the model's performance. The process continues until the best subset of predictors is identified.
3. Obtained the Final Optimized Model: the best logistic regression model is selected with only the most significant predictors.
4. Standardized Numeric Predictor Variables: a copy of the training dataset (df_scaled) was created to avoid modifying the original data. All numeric predictor variables (except Heart.Attack.Risk) were standardized using scale(). Standardization ensures that all numeric features have zero mean and unit variance, improving model convergence and stability.
5. Trained Logistic Regression on Selected Predictors: logistic regression model was built using the standardized dataset (df_scaled). Only the most relevant predictors (Cholesterol, Heart Rate, Triglycerides, Systolic, and Northern Hemisphere) were included.

Logistic Regression Accuracy

	Train	**Test**
Accuracy	55%	50%

- Overfitting is not observed in logistic regression model

<u>Steps to perform the logistic regression model on subset dataset:</u>

- Found the top 5 most correlated variables in each country subsets
- Trained the logistic regression with only the top 5 predictors

Logistic Regression Accuracy

Country	**Train Accuracy**	**Test Accuracy**
Italy	67%	52%
Japan	65%	50%

US	63%	49%
China	61%	52%

- Minor overfitting is observed since the train accuracy is higher than the test accuracy

Logistic Regression Analysis:

- Logistic regression model did not perform well in both the global data and subset data.
- Logistic regression assumes a linear relationship between the predictors and the log odds of the outcome. The relationship between heart attack risks and predictors is non-linear, this could be the reason why this model struggled to fit the data. The effect of one factor on heart attack risk may also depend on other factors. For example, for someone with high cholesterol, the effect of age on risk might be more pronounced than for someone with normal cholesterol.

Naive Bayesian

Steps to perform the naive bayes model on original dataset:

- Naive Bayes model was trained using only the most correlated variables: Cholesterol, Heart Rate, Triglycerides, Systolic, Northern Hemisphere

Naive Bayes Accuracy

	Train	**Test**
Accuracy	55%	50%

- Overfitting is not observed in naive bayes model

Steps to perform the naive bayes model on subset dataset:

- Found the top 5 most correlated variables in each country subsets
- Trained the naive bayes with only the top 5 predictors

Naive Bayes Accuracy

Country	**Train Accuracy**	**Test Accuracy**

Italy	65%	55%
Japan	66%	50%
US	65%	49%
China	60%	46%

- Minor overfitting is observed since the train accuracy is higher than the test accuracy

Naive Bayes Analysis:

- Naive Bayes model did not perform well in both the global data and subset data. It has a similar performance compared to Logistic Regression.
- Naive Bayes assumes that all predictors are independent given the outcome. This assumption can lead to poor performance if the predictors are actually correlated with each other. For example, if age, cholesterol, and smoking are correlated, Naive Bayes might fail to capture their combined effect on heart attack risk because it assumes that these features do not interact. This could be the reason why it performed poorly.

Random Forest

Steps to perform the random forest model on original dataset:

- Run the model on the full dataset. Then found the top important variables from the model.
- Using hyperparameter tuning to tune the number of variables randomly selected at each split to find the value that gives the best model performance based on the out-of-bag error estimate. Mtry = 3 has the lowest out-of-bag error estimate.
- Run the model on the selected features and use the best mtry (mtry = 3).

Random Forest Accuracy

	Train	**Test**
Accuracy	100%	52%

- Strong overfitting is observed

Steps to perform the naive bayes model on subset:

- Run the model with the best mtry value on the subset

Random Forest Accuracy

Country	Train Accuracy	Test Accuracy
Italy	100%	56%
Japan	100%	50%
US	100%	54%
China	100%	51%

- Strong overfitting is observed

Random Forest Analysis:

- Random Forest model is performing poorly in predicting heart attack risk and is showing signs of overfitting, even after tuning and selecting the best features.
- Random Forests can handle non-linear relationships, but if there are highly complex non-linear relationships between features and the outcome that are not captured by the trees, the model can still underperform or overfit. This could be the reason why it performed poorly.

Classification Tree

Steps to perform the classification tree model on original dataset:

- Use the rpart() function to train a classification tree model on the full dataset with Heart.Attack.Risk as the target variable.
- Identify the best cp value using cross-validation, prune the tree using prune().
- Predict on training and test sets and compute confusion matrices and accuracy.

Classification Tree Accuracy

	Train	Test
Accuracy	55%	51%

Steps to perform the classification tree model on subset dataset:

- Fit rpart() models for each country, Italy, United States, Japan, and China.
- Pruning was performed using the Complexity Parameter (CP) to avoid overfitting
- Predict on training and test sets and compute confusion matrices and accuracy.

Best CP Values and Accuracy on 4 Different countries on Classification Tree

Country	Best CP	Train Accuracy	Test Accuracy
Italy	0.0323	77%	52%
United States	0.0536	61%	45%
Japan	0.0692	67%	47%
China	0.0417	70%	38%

- The Classification Tree model performed better for Italy and Japan, with training accuracies exceeding 66%.
- The United States and China models have lower test accuracy, indicating potential overfitting.

k-NN

This model is presented in class. It performed well in country subsets but not in full dataset.

SVM

This model is presented in class. It performed well in full dataset but not in country subsets.

Conclusion

Recap of our presentation conclusion: SVM performed best for predicting worldwide heart attack risk, but with only 64% accuracy, we lack confidence in its reliability. However, k-NN showed strong performance in Italy and Japan. We observed a moderate linear correlation between certain factors and heart attack risk, and the accuracy of the models is moderately high.
Diving into our k-NN models, we identified the top three factors influencing heart attack risk in Italy and Japan. In Italy, the key factors suggest that heart attack prevention should focus on tackling obesity, improving healthcare access for low-income groups, and prioritizing high-risk individuals with a history of heart problems. In Japan, heart attack prevention should focus on promoting heart rate monitoring, improving dietary habits, and providing specialized care for individuals with a history of heart issues.
Some limitations of our project are Current dataset may not match the actual situation because it is generated from AI. Subsets of country data are small. Things that can improve our project would be to find larger, diverse real-time health data. And expand our analysis to predict heart attack risks in additional countries.

Other interesting conclusions:

- One possible reason for K-NN's success in predicting heart attack risks in subsetting data is its ability to handle non-linear relationships in the data. Unlike parametric models such as Logistic Regression, which assume linear relationships between predictors and outcomes, K-NN doesn't make assumptions about the underlying data structure, making it suitable for complex, non-linear patterns.

  Moreover, K-NN's sensitivity to local data structure gives it an edge when the data exhibits clusters or irregular decision boundaries. It excels in capturing patterns based on proximity, where heart attack risk may vary significantly between similar individuals. In contrast, other models, especially SVM and Random Forest, may struggle with complex decision boundaries or become prone to overfitting due to their reliance on specific kernels or tree depth without adequate tuning.

Another advantage of K-NN is its simplicity and lack of reliance on feature distributions. Unlike Naive Bayes, which assumes features are independent and follow specific distributions, K-NN directly computes distances between data points, making it robust in scenarios where feature relationships are difficult to model.

- It's very surprising that our random forest model is strongly overfitting even though we tuned the tree and selected the best features. We think this can be due to the dataset problem. And we might have to try out more features and model parameters to find the best model.