**TA comments:**

Team 30

Team Information (1/1): N/A

Objective/Problem (5/5): N/A

Dataset/Plan for Data (4/4): N/A

Approach/Methodology (7.5/8): \* Your dataset has several categorical variables, some with many categories. Will the curse of dimensionality be a problem? If so, how do you plan to address it - combine categories within a categorical variable, use alternate categorical encoding schemes, etc. \* Cross validation, by itself, is an approach to validate your model. Grid Search or random search are better techniques for hyper-parameter tuning.

Project Timeline/Planning (2/2): N/A

Total Points: (19.5/20) / (1.95/2)

Our approach:

* **implement a clustering algorithm to detect populations and assess the critical features for each population group** since most influential features that lead to heart disease are unlikely to be consistent across population groups:

# Data Preparation:

* **Basic data cleansing** (duplicate removal, handling of null values via interpolation or deletion).
* **Feature engineering** (one-hot encoding categorical variables – Race, Diabetic, and GenHealth; enumerating ordered categorical variables – AgeCategory; normalizing the range of continuous variables)
  + According to TA comments, we need to consider potential issues with dimensionality. One-hot encoding allows us to handle large number of categories in a variable by creating extra variables, but it’s computationally heavy. We could 1) reduce categories within each variable (ex. 10 categories into 2: 1,2, …, 10 🡪 1-5 & 6-10); 2) use alternate categorical encoding schemes, such as dummy encoding or label encoding. We could perhaps implement principal component analysis, too.
  + I think the following links are good places to start:
    - <https://annahava.medium.com/too-many-categories-how-to-deal-with-categorical-features-of-high-cardinality-d4563cfe62d6>
    - <https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f>
* May need to split data into (training/test) or (training/validation/test) sets for evaluating model performance later

# Data Exploration

* Scatterplot
* Histogram
* Correlation matrices 🡪 check for multicollinearity graphically, or via VIF
* Rudimentary modeling using lm(), summary() & plot()
  + **investigate variables for normality and variability of variance** to check if we need any transformations
  + **outlier detection using Cook’s distance**
* Use ANOVA to evaluate the significance of our features across community groups to detect if different features are significant for different communities of people.

# Modeling

* **stepwise logistic regression** model as a method of feature selection
  + Once the features are down-selected, apply clustering via k-Nearest Neighbor to detect unique groups within data. Other clustering algorithms will be considered if k-NN produces poor results
* **Compare performance of models**: logistic regression, XGBoost, and AdaBoost
  + Use confusion matrix (accuracy, precision, recall, and emphasis on f1-score)
    - How conservative should our classification threshold be? Is it more important to minimize false negatives at the cost of classifying more people as “at-risk?”
  + Create ROC curve & calculate AUC
    - We could stick to classification threshold of p=0.5, but if we have time, we could play around with other values to see if we get better accuracy/precision/recall according to how conservative we want to be in our predictions
* F1-score will also be the metric we utilize during hyper-parameter tuning with cross-validation.
* TA comments: Cross validation, by itself, is an approach to validate your model. Grid Search or random search are better techniques for hyper-parameter tuning

# Interpretation, Pitfalls, Suggested Course of Action, etc.

* I think each person can do their own interpretation/analysis of potential errors/suggestions for next steps, and people can give feedback on each other’s work as we go