Data Analysis and Preprocessing

- No missing values identified; therefore, no processing needed in this aspect.
- Any sample points more than 3.5 standard deviations above or below the mean are removed from the dataset. We do this to prevent unlikely outliers potentially due to measurement errors from skewing the model results. In future iterations, it may make sense to test model performance both with the outliers included and without the outliers included.

Model Development:

Ultimately, three models were tested:

1. Logistic Regression
2. Decision Trees
3. Random Forest

Throughout our analysis, one of our areas of focus was to avoid unnecessary complexity and overengineering. Thus, we began our analysis with a simple classification model - that is, the logistic regression. Then, we measured its performance. Since its performance was not up to par, we gradually began testing more advanced models until we were able to meet our performance standards. After the logistic regression, we implemented a decision tree model and then a random forest model. Ultimately, the most successful model, the random forest, was able to reach 98% classification accuracy. Measured on 10-fold cross-validation, it was able to reach about 91% accuracy. Below, we outline our hyperparameter choices for the 3 models:

- Logistic Regression
  - StandardScaler() - Used because logistic regression is sensitive to the scale of input features. Standardizing ensures all features contribute proportionally to the model.
  - Multi_class = 'ovr' - This sets "One-vs-Rest" strategy, which is suitable because we have multiple classes. It treats each class as a binary problem (one class vs. all others).
  - Max_iter = 1000 - Increased from the default (100) to give the model more iterations to converge.
- Decision Trees
  - Max_depth = 5 - Limits the tree to 5 levels deep.
  - Min_samples_split = 5 - Requires at least 5 samples in a node before it can be split further. Can prevent overfitting by ensuring splits are statistically meaningful.
  - Min_samples_leaf = 2 - Requires each leaf to have at least 2 samples.

    - Note that we do not use StandardScaler() since decision trees do not need feature scaling.
- Random Forest
    - n_estimators = 100 - Creates 100 different decision trees. Scikitlearn default that balances computational cost with model robustness.
    - min_samples_split=2 - We allow splits to occur with as few as 2 samples. This is a scikitlearn default that emphasizes model robustness and generalizability.
    - min_samples_leaf=1 - We choose to allow leaves with 1 sample. This is a suitable selection for random forests, but not for decision trees.