Group Members: Christopher Carlsson, Julia Ferrante, Ray Valenzuela, Catherine Eng

- **What algorithms will be explored and why?**

We have decided to explore k-nn, decision trees classifier/regression, and random-forest classifier/regression. All of these algorithms are popular for supervised machine-learning tasks, and they can all be used for both classification and regression tasks. Below is a comparison of the three:

K-Nearest Neighbors (K-nn) make predictions based on the k-nearest data points in the training set. This algorithm was chosen because it is easy to understand and implement. It is particularly efficient when the data is not very complex and the goal is to have a simple model that makes good predictions.

Decision Tree is based on a tree-like structure of if/else statements. It is good at handling both categorical and numeric data, which is why we chose it. One key downside is that it is prone to overfitting. The advantage is that it works great with smaller data sets.

Random Forest Classifier combines many decision trees to make a prediction. It was chosen because it is superior to k-nn and decision trees when handling missing data, and outliers, and when the data set is large and complex. A downside is that this algorithm is computationally expensive, which is why the other two algorithms are preferred in handling smaller data sets.

- **How will the model be trained and evaluated?**

All three models will be trained by performing a train test split, with a 0.30 test size which means 70% of the data fed into the model will be used for training and 30% will be used for a test set. This ratio of training to test set can be modified depending on the performance of the models. The K-nearest neighbor model will be evaluated by generating a classification report and assessing the model's accuracy and precision for each group. The decision tree classifier and random forest classifiers will also be assessed by calculating the models' accuracy and determining if it is acceptable or not. Accuracy can be used for evaluation since these models are performing classification rather than regression.

- **What parameters will be tuned (i.e. hyper-parameter tuning)?**

Grid search can be used to find the best values for each hyper-parameter. For the K-nn model, the parameters that will be tuned are k (the number of nearest neighbors), the metric to determine which distance calculation method will be used, and leaf_size, which can affect the speed. The decision tree can be tuned using the hyper-parameters of max_depth (dept of tree, can cause overfitting/underfitting), max_features (number of features considered for best split, can help with overfitting and computational intensity), min_samples_split (minimum number of samples required to split an internal node, can help control overfitting), and min_samples_leaf (min number of samples required to form a leaf, can smooth the data). The random forest classifier will be tuned similarly to the decision tree model, using n_estimators (the number of trees in the forest), max_depth, max_features, min_samples_split, and min_samples_leaf.

- **What is the measure of accuracy that you expect from this model and why?**

The average accuracy of a knn model is just above 70% accuracy. Somewhere in this same range is what we expect for our model as well. There is a large variety of features and a lot of data to use which should push our model closer to this average. On the other hand, however, too many features may cause overfitting of the model. Building our models carefully will ultimately help us push our accuracy to its maximum potential.