**Deliverable 1**

A Portuguese banking institution used data obtained through its marketing campaign to predict whether clients are likely to subscribe to a term deposit. The bank’s marketing campaigns were centered on making direct phone calls. The data acquired shows a variety of information pertaining to the potential clients, including such metrics as age, job, marital status, education, and loan history. They also recorded information regarding the specifics of each phone call and marketing representative, including the duration of the call, the day/time of the call, the number of times the client has been called and whether their previous call was successful.

We must begin by conducting further analysis to accurately predict whether each client will subscribe to their term deposit. Our project will be considered successful if we can accurately predict which set of conditions will lead to a client subscribing to a term deposit. We must construct a model that uncovers the precise groups of individuals and the times they are likely to subscribe. In doing so, we can present these findings to the Portuguese banking institution so they can focus their marketing efforts in a more efficient manner.

In terms of cleanliness, our main issue was the ‘unknown’ values were not read as null and required correction. Several of the columns delivered as ‘object types’ with values of either yes or no, so we decided to change these datatypes to Boolean. We replaced missing values on several of the columns with the most common value, and all instances of yes and no with 0 and 1. Numerous column headers were problematic and were updated for clarity. We our grouped a lot of our categorical data together in preparation for creating dummy variables, which we plan on using to create either a decision tree or random forest. We decided to drop the “default” column because only 3 of the 40,000 values were yes. We also dropped the “pdays” column because they decided to put the value ‘999’ for every new customer (which skewed the data) and it already felt redundant due to the similar ‘campaign’ and ‘previous’ columns.

We assumed that single people, people with higher incomes, people with personal loans and people with home loans were more likely to say yes. After analyzing the ‘campaign’, ‘previous’ and ‘poutcome’ columns we assume that clients will be more likely to say yes if they did so the first time and will be less likely to respond or say yes if they’ve already been contacted several times beforehand. Duration factors in too heavily because longer phone calls will almost always lead to yes, so we will neglect this column during our analysis.

**Deliverable 2**

After creating the train\_test\_split and running a random forest, we struggled to get an R-Squared value above 0.2. After some discussion, we decided to go back and create grouped values for duration and age so that we could have more predictors. After creating roughly five bins for each duration and age and recreating our dummy variables we brought our predictor total up to 38. We also decided to normalize our economic indicators to see if it would have any impact on the model. We finally realized we had been using the incorrect random forest for our dataset.

Before using the correct random forest, we decided to start with a decision tree to have a baseline for comparison. We created our first decision tree using a max depth of 5. Our decision tree produced an accuracy of 0.91. We checked the confusion matrix to further explore the performance. Our model accurately predicted our “yes” 405 times. We had 535 false positives. We knew that the “yes” predictions would be difficult given the match smaller frequency when compared to “no”. This is not a perfect model, but it does not seem bad. Our biggest concern with the model is that it is overfitting the test data.

Next, we decided to try the random forest classifier model, which would lower the chances of overfitting and likely provide a more reliable model. We chose 200 estimators, max depth of 15, and bootstrap equal to true. This model produced an accuracy of 0.90. It predicted “yes” 347 times, compared to 593 false positives. Though the performance of our random forest model seems to be worse than the decision tree model, we believe it will be more reliable due to the overfitting issues of our decision tree.

We then attempted to fine tune our random forest model using the randomized search cross-validation method with the hyperparameters bootstrap, max\_depth, min\_samples\_split, min\_samples\_leaf, max\_features and n\_estimators. The main reason we chose the randomized search method over the grid search was the faster performance. The grid search seemed to take much longer, which is logical based on the way it works. Our cross-validation recommended parameters were n\_estimators=100, max\_depth=6, min\_samples\_split=14, min\_samples\_leaf=5, max\_features=28, bootstrap=True. We entered the recommended parameters into our random forest model and ran the model again. This model produced an accuracy of 0.91. It produced 375 correct “yes” classifications compared to 565 false positives. Both of these metrics were an improvement to our previous random forest model, so they hyper tuning was a success. We viewed the classification report to further explore our results. This showed us that our “yes” predictions were not terrible, but not great either. We tried changing some of the features used in our model, but we did not have any luck in creating a better model.

We created a bar graph and list of all important features using the feature\_importances\_ function and learned that the nr\_employed (number of employees quarterly indicator) and bin\_duration\_LongestCall were by far the most important predictors at 0.32 and 0.27. The third most important predictor only had an importance of 0.10.

We were hoping several features would prove to be better predictors than the rest, but this turned out not to be the case. None of our variables turned out to be great predictors and only a small amount of people from the original dataset said “yes” to subscribing to a term deposit in the first place. These two factors hold our model back from reaching its full potential, but we believe that the model is strong given the information we had to work with.