This document is a technical report for the data science challenge for Gannett written by Colin O'Callaghan. Please refer to the accompanying jupyter notebook titled "ds_challenge_response.ipynb"

The data was presented in a csv file with a seperate text file containing the column headers, or feature headers. The data was read into a Python pandas module, this data format was chosen for the many built-in data science functions and compatibility with Scikit-learn functions. An initial exploration of the data was performed to understand the different data-types contained in the set. There are 10 columns in the data-set, 8 numerical and 2 contained different classes. There are 10,000 points in the data-set, 8000 with churn=1, 2000 with churn=0. Several interesting patterns were observed in the data and can be seen in the jupyter notebook. One feature, "CPL_wrt_self" contained over a thousand NAN values, here these were replaced with the mean value of this feature to enable analysis of these entries. With hindsight the median value of 0 should have been used as a more reflective replacement of the NaNs as it is not as sensitive to the outliers present. Each pair of numerical features were plot against each other, coloring the data points differently for the two churn numbers. No obvious clusterings of churn numbers were identified in this plot.

It was decided to apply a classifier to the data. To do this, the data needed to be transformed as the features must be numerical. The client states and business categories were hot-encoded, a best practice approach to turning classes into numerical data. The nan values in the CPL_wrt_self feature were replaced with their mean values and an additional column was created that signified if they were originally a Nan or not. This was done so as to not remove information from the data. The data was then split into train/test data. To look for a good ratio for train to test data, an off-the-shelf random forest classifier was used to show that an 80% / 20% ratio would work. Kfold could have been used for training, however the data-set is small with relatively few churn = 1 samples, it was decided to use the train/test data sets as it would provide more churn=1 samples to the classifiers during fitting. With a bigger sample size Kfold would be more advisable.

Several off-the-shelf classifiers were applied to the data in order to determine which model may be most suitable. A more simple model would be logistic regression which did not perform very well. However by normalizing the numerical data this model could perform better. A random forest classifier was show to have the highest mean accuracy of 83.15% (correctly predicted the churn number of data 83.15% of time). This is higher than the naive approach of guessing each churn number is 0, which has an accuracy of 81.6% on test data. This naive approach does not offer any insight to the system. The

random forest does suffer from over-fitting as it perfectly classifies the training data. This is an issue that should be addressed in the future.

An excellent property of random forests is that the relative importance of each feature can be estimated. The “CPL_wrt_self” was shown to be the most important feature, duration followed slightly behind that. The individual locations and business categories were shown to not be as important as the 8 other features. By removing unimportant features from classifiers their performance can be increased, this was shown not to be the case for this data-set.

To better quantify the performance of the random forest, the confusion matrix was examined. Here the model was shown to have a very low recall rate (20%) and a high precision rate (73%). Depending on the application of the model to the business needs, either a high recall or high precision is more desirable. This can be achieved in the model by changing the threshold value at which the model predicts a churn of 1. However, by lowering the threshold value the precision can drop, i.e. the model gives false positives. This is summarized in the precision-recall and roc curves of the model.

My final conclusion is that the current random forest model does classify the churn number of the test data well. Several improvements to the current model could be made. The CPL_wrt_self should have the NAN values replaced with the median values instead. A comprehensive examination of the parameter space used to define the random forest should be carried out, code to do this is presented at the bottom of the jupyter notebook. However this process is computationally expensive. The numerical data could be normalized, in particular the outliers present in some features may have unwanted impact on the final model. The test/train data set should also be split conserving the ratio of churn values.