Introduction: Summary and problem definition for management

Assignment 2 for this class is an exercise in identifying the best classifiers and machine-learning methods to model consumer behaviors. The example is taken from Thomas Miller's book 'Marketing Data Science'[1] and uses a data set from exercise C.3 – the 'Bank Marketing Study'. The purpose of the analysis is to do an initial exploration of a small data set gathered from a telemarketing campaign at a bank, apply various machine-learning techniques to develop a range of predictive models, and then validate which models were most successful at predicting customer behavior. The methods developed, and recommendations for best 'classifiers' for prediction will be useful to pre-screen customer data for any future marketing campaign by focusing resources on the individuals most likely to make a purchase.

Discussion of the research design, measurement & statistical methods

The data set provided is a csv file with 4521 rows of data. Each row is a customer record that includes 17 variables specific to the client's demographics, their assets with the bank, when and how they were contacted, how frequently they were contacted and if they purchased a term deposit. The data is a mix of categorical variables, continuous variables and binary variables – examples here show the variable titles of the variable columns in quotes (''). Categorical variables include classifiers such as 'job', 'marital' status, 'education' level, and 'poutcome' - the outcome of the previous call. Continuous variable examples include the 'age' of the client, their 'balance' in their bank account, and the 'duration' of the previous call in seconds. Binary variables are all yes/no answers to questions like: does the customer have any credit in 'default', do they have a 'housing' loan, do they have a personal 'loan', and if the outcome of the call was a positive 'response'.

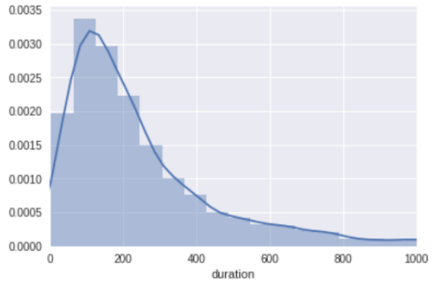
88% of the time, the response to the marketing call was 'no', so the objective of this analysis is to understand which variables had the most influence over the outcome. After converting all binary variables ('No' = 0/ 'Yes' = 1), and converting categorical variables into 'dummy' variables, I grouped the variables by the response outcome, and then sorted by the mean difference between positive response an negative response. The result was that the five variables shown to the right had the most distance between groups that responded positively and those who did not.

The top 5 variables with the most Δ between means when grouped by 'No' response and 'Yes' response.

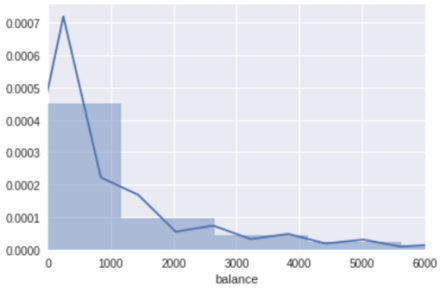
response	index	0	1	mean_diff
0	duration	226.34750	552.742802	326.395302
1	balance	1403.21175	1571.955854	168.744104
2	pdays	36.00600	68.639155	32.633155
3	age	40.99800	42.491363	1.493363
4	previous	0.47125	1.090211	0.618961

Below are some initial distribution views of the top 3, which we will discover have a role to play in the model analysis:

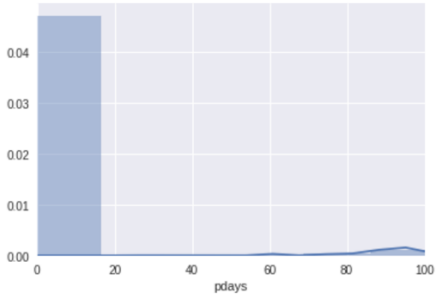
'duration'
the length of previous call in seconds



'balance'
Average yearly balance in Euro



'pdays'
# of days since last contact



Overview of the programming work

In this analysis, I ran three methods of machine learning model development: 'nearest neighbor' analysis, Naïve Bayes, and logistic regression using a multiple k-fold method. For the last two methods I compare the prediction validity of test models using multiple folds and using two different variable groups to determine which has the best proven performance. To achieve

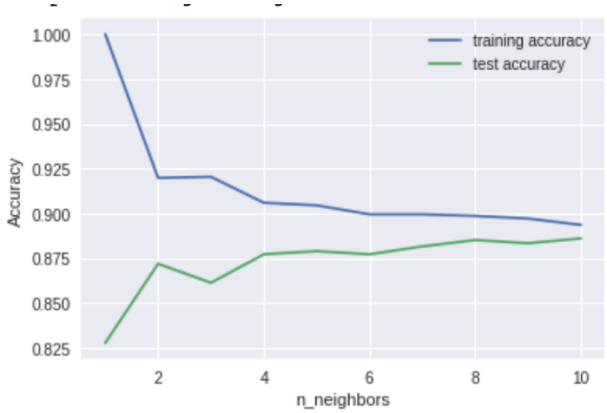
[1] Miller, T. W. 2015. *Marketing Data Science: Modeling Techniques in Predictive Analytics with R and Python.*

this, I first had to build a training data set and a test data set, so we could first build the model and then 'validate' its predictive capability. The function train_test_split' within the sklearn.model_selection library performed this function easily.

Model 1:	**Model 2:**	**Model 3:**
Evaluation using 'KNeighborsClassifier' **variables unknown - test performance*	Naïve Bayes vs. Logistic Regression (1) **suggested = 'default', 'housing' & 'loan'*	Naïve Bayes vs. Logistic Regression (2) *selected = 'duration', 'balance', & 'pdays'

The **first model** that I examined used the sklearn.neighbors library to determine what was the ideal number of classifiers to balance performance across both the training and the test data sets. As the number of 'neighbors' increases the accuracy of the training set comes down and the accuracy of the test set goes up. As the general rule is that we want to minimize model complexity, the 'right' number of classifiers may be in the range of 4 to 6, where test model accuracy goes above .875, but we are likely not in a range where we are 'over-fitting' the model.

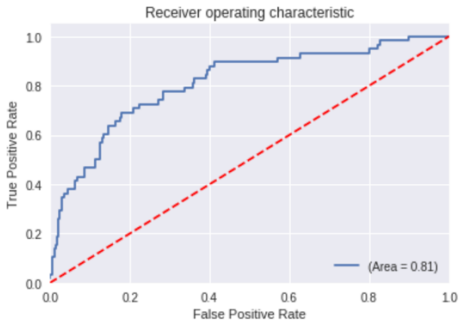
The nearest neighbor model gives a good idea of the predictive potential against the test data set, but the text book suggests it may not be the best choice in terms of model practicality. Therefore, I will use this as an example of performance 'potential' as I look to model 2 and 3, which examines both the Naïve Bayes and logistic regression methods averaged across 10 'folds'.



Model 1 performance against # of classifiers

In the **second model**, we take the 'jump start' code which suggested binary variables of 'default', 'housing', and 'loan' as the primary predictors against the response. In the initial data exploration above, these variables had a low difference between means when grouped by response, so they appear to have much less influence than the variables I suggest for Model 3. It is no surprise that the prediction validation scores come out fairly low when run against the 10 fold cross validation. Both the NB and logistic regression methods result in a disappointing .61 area in the ROC curve, meaning that they perform only slightly better than picking response variables at random.

10 fold cross validation outcome – Model 3




```
------------------------------------------------
Average results from 10-fold cross-validation

Method                 Area under ROC Curve
Naive_Bayes            0.613884
Logistic_Regression    0.825396
dtype: float64
```

In the **third model**, I decide to use the variables for 'duration', 'balance' and 'pdays' instead of the recommended ones. The initial data exploration above suggested that these variables will play a more significant role in predicting purchasing behaviors, and the validation of the test data using the logistic regression model suggests the same.

In this case, the Naïve Bayes model did not average much better than Model 2, with an area under the ROC curve of .61 still. However, the logistic regression model performed much better with an average of .825. In one of the 10 folds, the test results showed close to .88, which is in line with the nearest neighbor validation average that we saw in Model 1.

Review of results with recommendations

The recommendation, therefore, would be that the team at the bank consider using a logistic regression model using the most highly influential predictive variables. Given more time, I would go a few steps further to explore additional variables, such as 'age' and 'previous' outcome.

I would also recommend that the data set for analysis be improved, as the data is limited to 4500 instances. One method to improve on the robustness of the data set may be to run a bootstrap technique of the training data to randomly sample more responses to train the model.

Note:

A shared version of the python notebook used to create this analysis can be found at Google CoLab:
https://colab.research.google.com/drive/19xvGRTSm2rcy7nDxZs1bfQ34J4Eh1LGO