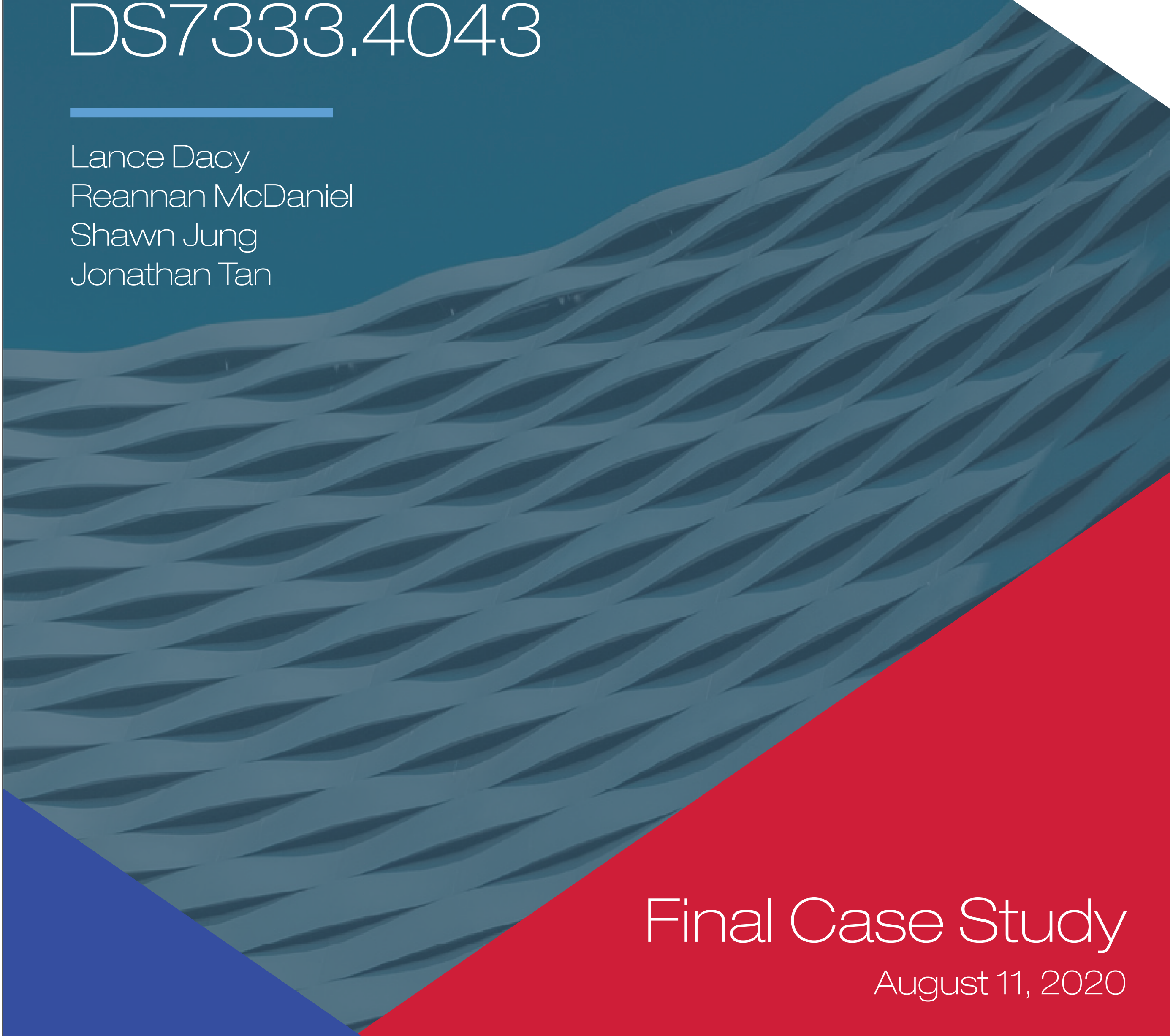
SMU®

Data Science Quantifying the World DS7333.4043

Lance Dacy
Reannan McDaniel
Shawn Jung
Jonathan Tan



Final Case Study

August 11, 2020

Table of Contents

Introduction

Thank you first and foremost for the opportunity interview with your firm. In the ever increasing discipline of data science it is increasingly more difficult to find the right person with the tools of data science along with the skill of analysis and decision-making or even problem solving.

At the onset of this resume being selected to go forward, it is with great respect for the hiring process that the focus remains on the following elements when working through the problem stated:

- Best Practices
- Estimate your model performance on the data provided
- Providing analysis on the thought process along the way

This paper will serve as a summary for the experience with the data provided, the analysis to reach a decision on feature selection, experimentation with various models, and finally choosing a model that aligns with the following business objective (minimizing cost of errors):

- Each False Positive costs $10
- Each False Negative costs $500
- True Positives and True Negatives cost $0

While reviewing technology choices and skills along the way are important to ensure the candidate has a grasp on an approach, this paper serves as a business summary. More details can be found in the GitHub code repository. Generally, there is a high-level framework that most data scientists take when provided a question of interest:

- Step 1: Frame the problem.
- Step 2: Collect the raw data needed
- Step 3: Process the data for analysis
- Step 4: Explore the data
- Step 5: Perform in-depth analysis
- Step 6: Communicate results of the analysis

This experiment will take a similar approach above; breaking down each Step into a table of contents.

Data

Naturally, data is the most important starting place in the approach to analyze a problem. Data was provided for this task, as such, the steps to view, clean, and analyze the data will be outlined sequentially.

Data Source

For the question of interest, an "UNKNOWN" dataset was provided by the hiring manager. Leaving the dataset as "UNKNOWN" ensures no one candidate has a head start in domain expertise. The first step is to approach the data to gain as clear of an understanding as possible given the unknown domain. This usually starts with an EDA process (exploratory data analysis) that can be different based on the features, size, and relevancy of the data.

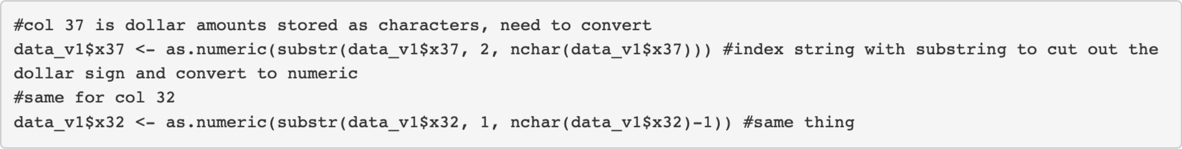
Exploratory Data Analysis

The EDA task was completed using R Studio. The understanding of the tasks is to classify column "y" using features x0-x49 to minimize costs. Using a skim function; a summary of the data frame can initially be reviewed to determine next steps.

Missing data is one of the first issues a data scientist will likely tackle after gaining access to the data. Initial steps include determining variable means and variable histograms. This dataset appears to be missing data under the triple digit mark which is very few compared to the 160,000 rows of available data. Of note is that the average value of "y" is 0.401, indicating an estimate of 40% of the rows having y=1.

In addition, a review of rows that include "NA" was completed. There is roughly 1,500 rows that are removed that include "NA." This makes up for 0.9% of the data. There certainly are techniques that could be employed to determine imputation standards for this dataset. At this point it is of note to review how these missing data may affect the models once those are explored in detail. In practice, simply taking the median value and applying it for missing values arrives at similar results to the baseline in a model with missing data. This encourages the data scientist to avoid mechanically omitting missing-data records for the sake of saving time and using a few forms of imputation to strengthen the model. At this stage however, the data is simply being reviewed and assessed.

Attributes x37 and x32 are dollar figures and includes a "$" in the cell which is interpreted as a character. This field needs to be converted into a numeric format. A function to index the string with a substring was generated to remove the dollar sign and convert to numeric for both attributes.

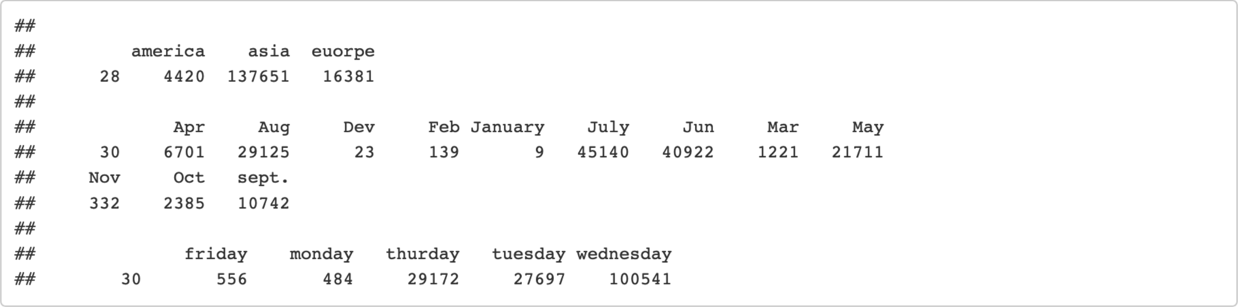


```
#col 37 is dollar amounts stored as characters, need to convert
data_v1$x37 <- as.numeric(substr(data_v1$x37, 2, nchar(data_v1$x37))) #index string with substring to cut out the
dollar sign and convert to numeric
#same for col 32
data_v1$x32 <- as.numeric(substr(data_v1$x32, 1, nchar(data_v1$x32)-1)) #same thing
```

In addition to the cleaning of the characters for a numeric field, a function was employed to remove the NA which accounted for another 70 rows that were removed. In this step it was determined that the 70 rows were negligible in the overall dataset. Again, imputing functions could be explored once a model is chosen.

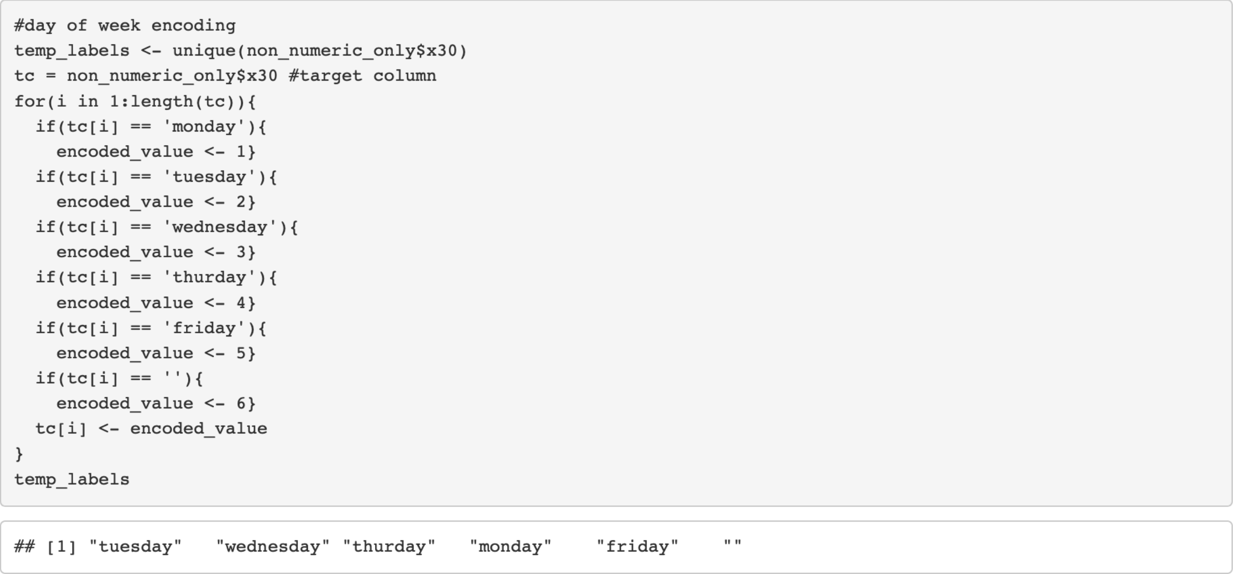
An encoding feature is employed due to a few other variables that require categorical manipulation. The sheer number of categorical variables in each non-numeric column is of interest. Variable x24 appears to be a region or continent and includes “america,” “asia,” and “europe.” Respectively there are

- 4,420 records for “america”
- 137,651 records for “asia”
- 16,381 records for “europe”
- 28 records that were blank.



```
##
##         america    asia  euorpe
##      28    4420  137651   16381
##
##             Apr     Aug     Dev     Feb January    July     Jun     Mar     May
##      30    6701   29125      23     139       9   45140   40922    1221   21711
##     Nov     Oct   sept.
##     332    2385   10742
##
##               friday    monday   thurday   tuesday wednesday
##        30       556       484     29172     27697    100541
```

The same exercise was done for attributes x29 (month) and x30 (day of the week) excluding Saturday and Sunday. Given the domain knowledge of how months, days, and calendar items are categorized, it is important that these remain consistent throughout the data.



```
#day of week encoding
temp_labels <- unique(non_numeric_only$x30)
tc = non_numeric_only$x30 #target column
for(i in 1:length(tc)){
  if(tc[i] == 'monday'){
    encoded_value <- 1}
  if(tc[i] == 'tuesday'){
    encoded_value <- 2}
  if(tc[i] == 'wednesday'){
    encoded_value <- 3}
  if(tc[i] == 'thurday'){
    encoded_value <- 4}
  if(tc[i] == 'friday'){
    encoded_value <- 5}
  if(tc[i] == ''){
    encoded_value <- 6}
  tc[i] <- encoded_value
}
temp_labels
```

```
## [1] "tuesday"   "wednesday" "thurday"   "monday"    "friday"    ""
```

Numeric values were assigned to each region:

- “america” = 1
- “asia” = 2
- “europe” = 3
- blank = 4.

A “region_numeric” data frame is created for later use in the model.

Monthly encoding is also performed to convert variable x29 (months) into numeric form. A ”month_numeric” data frame is created for later use in the model building. Finally the same process is completed for attribute x30 (day of week). Similarly, a “day of the week_numeric” data frame is created to use in the model building step. Each of these categorical changes were vetted by backing into the data using a table function to produce the output before and after the conversion for repeatability in the data (if new data was provided later).

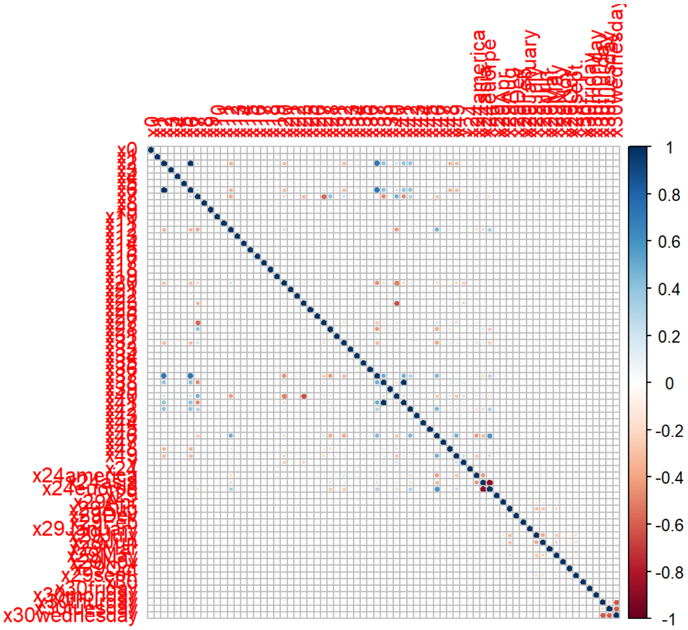
Dummy variables were then created for each of the categorical variables. This required four new variables for region, 13 new variables for month, and 5 new variables for day of the week.

Correlation Analysis

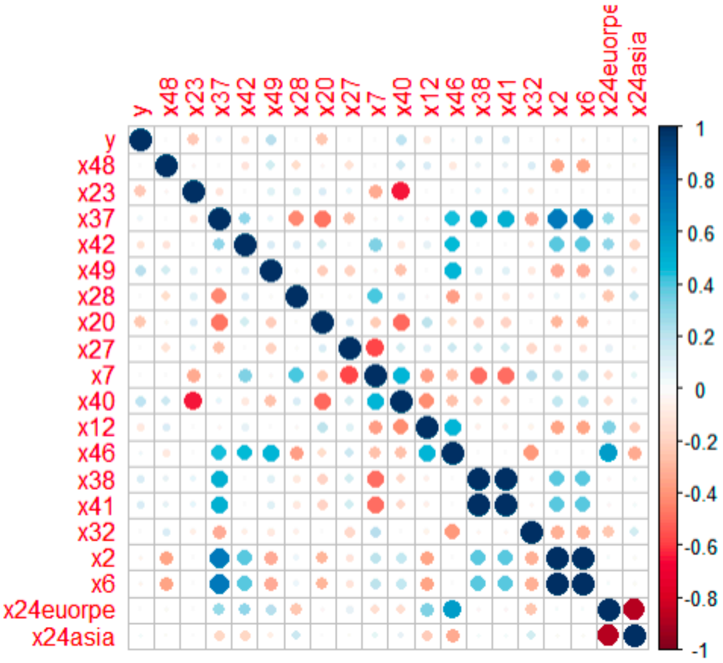
After cleaning the data, the next step is to evaluate correlation between the data to narrow down a model approach for prediction. Attributes x24, x29 and x30 are region, month and day of the week (weekday only) respectively.

Each of these went through one-hot encoding which generated new dummy variables for binary values in each category. About 0.9% of the is data missing with minimal value from imputing; therefore these rows values were removed from the data set. The values were imputed into the new data frame including the new dummy variables in order to generate correlation analysis.

According to the correlation plot of all the variables, there are appears to be strong negative correlations between Wednesday and Tuesday, as well as Asia and Europe. There does not appear to be strong correlation among any of the individual categorical variables and target y.



Reducing the correlation matrix down to the top 20 variables provides a much more clear view of the relationship.



Model Summary

Myriad tools exists today to help data scientist in their quest to answer predictive questions for their clients. Getting the data in the right shape is the first step towards experimenting with various models and assessing their performance against the metric that the client is requesting.

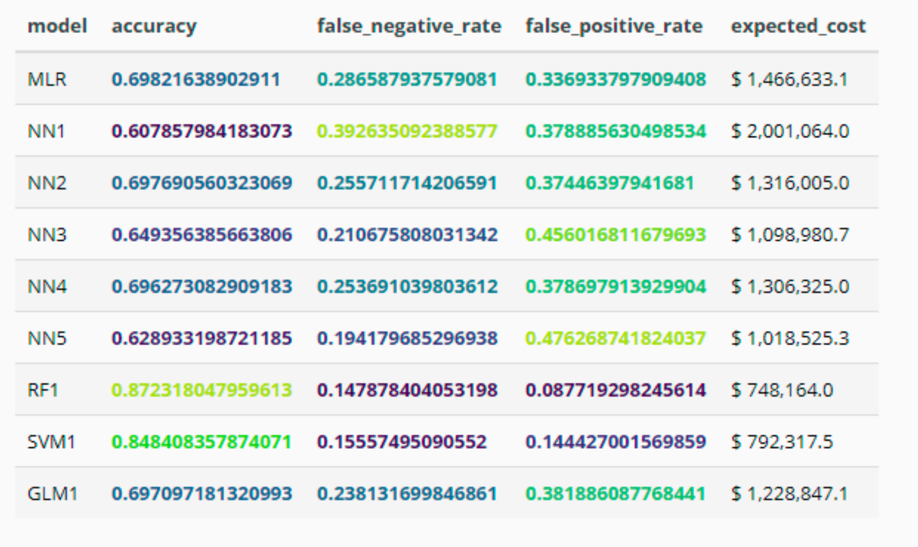
Models can be broken down into supervised or unsupervised learning problems. If the data is labeled, it's a supervised learning problem. If it's unlabeled with the purpose of finding structure, it's an unsupervised learning problem. If the solution implies to optimize an objective function by interacting with an environment, it's a reinforcement learning problem.

Based on the data presented and the initial analysis; there were a total of 7 various models that were used to draw closer to the question of interest and predict the results within the threshold / parameters instructed:

- Model 1: Single Layer Neural Net
- Model 2: NN BackPropMomentum
- Model 3: NN w/BackpropMomentum and TANH
- Model 4: 3 layer NN w/BackpropMomentum and tanH
- Model 6 (SVM)
- Model 7 (GLM)

A more detailed analysis of these models can be found in the GitHub Repository.

In summary, the results against the expected cost provided by the client were compared amongst each other. The data in the figure below was used to enable a high level comparison of initial performance across the models (higher values are indicated as brighter):

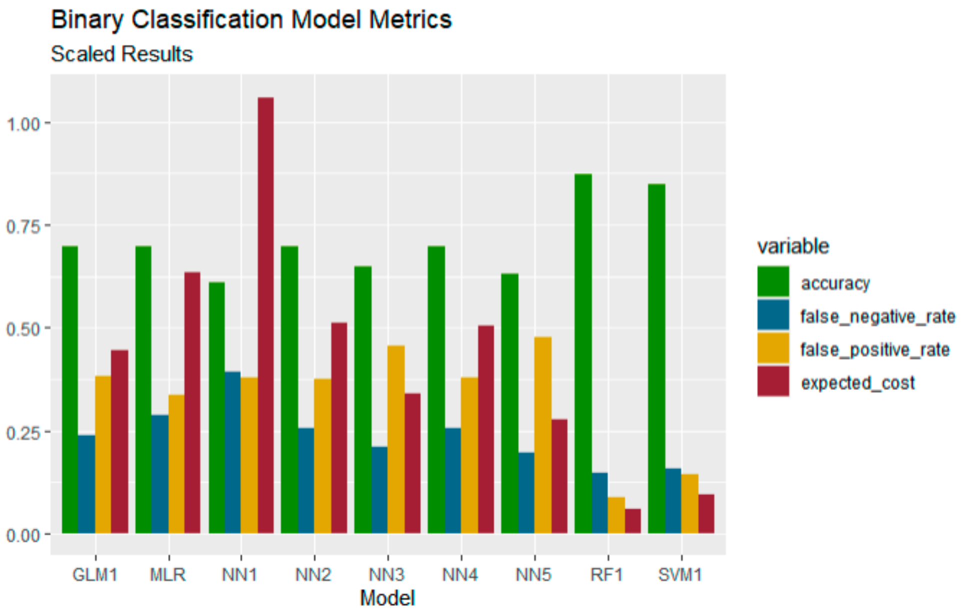


model	accuracy	false_negative_rate	false_positive_rate	expected_cost
MLR	0.69821638902911	0.286587937579081	0.336933797909408	$ 1,466,633.1
NN1	0.607857984183073	0.392635092388577	0.378885630498534	$ 2,001,064.0
NN2	0.697690560323069	0.255711714206591	0.37446397941681	$ 1,316,005.0
NN3	0.649356385663806	0.210675808031342	0.456016811679693	$ 1,098,980.7
NN4	0.696273082909183	0.253691039803612	0.378697913929904	$ 1,306,325.0
NN5	0.628933198721185	0.194179685296938	0.476268741824037	$ 1,018,525.3
RF1	0.872318047959613	0.147878404053198	0.087719298245614	$ 748,164.0
SVM1	0.848408357874071	0.15557495090552	0.144427001569859	$ 792,317.5
GLM1	0.697097181320993	0.23813169984861	0.381886087768441	$ 1,228,847.1

The random forest model (RF1) has high accuracy, low false positive and false negative rates, and the lowest estimated cost on a theoretical sample of 10,000 new rows.

Due to time and hardware constraints, not every model was fed the same number of rows from the original dataset. To create a meaningful comparison, ratios of various metrics were used instead of flat numbers. This would give the appearance of favorable results to models that had a lower number input cases.

Accuracy is from the number of correctly classified cases / total number of cases. False negative and false positive rates, or rate of type 2 and type 1 error, are the ratios of each error type relative to the total number of rows fed into the models.



Expected costs are the expected numbers of type 2 and type 1 errors, multiplied by 10,000 theoretical rows of new data. Then each is multiplied by 500 and 10 respectively to get an estimated cost per 10,000 rows of new data.

Note that expected_costs are scaled from 0.1 to 1.1 in order to fit the cost ratios on the same scale as the other metrics. The random forest model once again performs best in this scenario.

Random Forest Model (RF1)

At the onset of the problem for this exercise, there are plenty of approaches that can be used. Random forest however tends to be a flexible, easy to use machine learning algorithm that produces a great result most of the time.

It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks). It was no surprise that out of the 7 models assessed, this one would win out for its performance, scalability, and cost reduction of the question of interest.

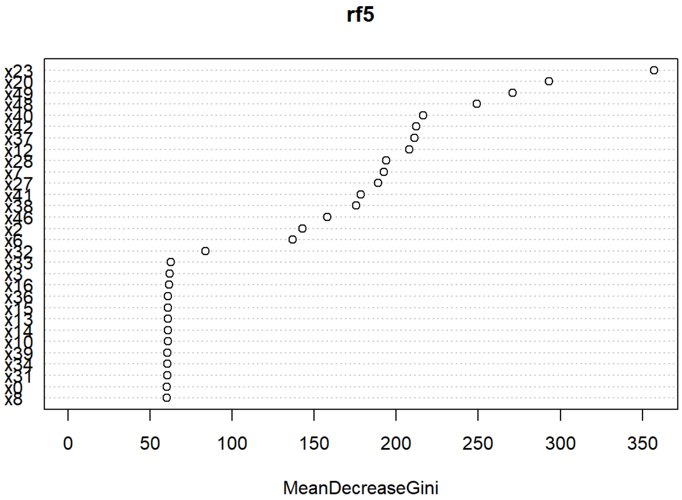
As mentioned in the EDA and Correlation Analysis there are variables at the onset that seem to help in tuning the model. Random Forest can also employ a feature importance selection that helps reduce the features even further to narrow down the most important ones for prediction.

Results

Variable importance from a Random Forest model is determined by looking at the prediction error rate for each variable at each 'fork' in the decision tree. The importance of that variable is calculated using the ratio of the number of times that variable lead to a desired classification.

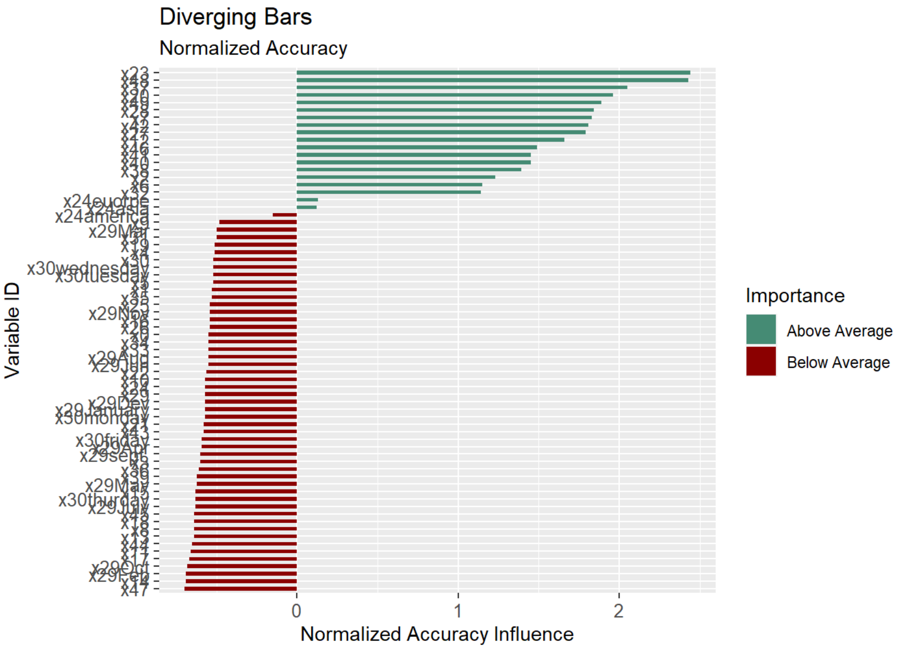
Mean Decrease in Accuracy is a measure of how much the accuracy of a classification drops when that variable is excluded from the tree. Therefore variables with higher mean decrease accuracy are more important to the model.

A table of variable importance shows that x23, x48, and x20 were the strongest indicators/ most important variables in making a '0' classification.



In addition to the feature importance graphic above, a diverging bar chart can also help deduce which feature might affect the results. A diverging bar chart is a bar chart that has the marks for some dimension members pointing up or right, and the marks for other dimension members pointing in the opposite direction (down or left, respectively).

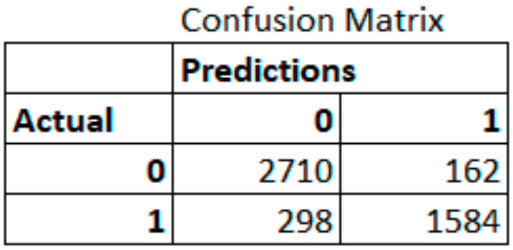
The chart below shows relative importance and weight for each of the variables with regards to accuracy and classification as a '0' or '1'. Most of the variables aren't significant to the accuracy rating of this model.



Conclusion

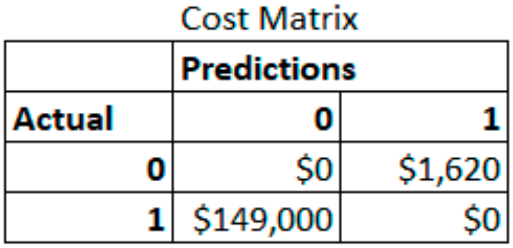
Running the model tuned with the same random forest but selecting only the variables that had a positive influence on the accuracy rating proved to be the most cost effective model. It was therefore selected to create the final result to the question of interest.

Given the confusion matrix below for the final model, it is determined that false positives at 162 will cost the client $1,620 (162 X $10). The cost for false negatives at 298 will cost the client $149,000 (298 X $500). Therefore this model produces a total cost of errors at $150,620 which is significantly lowers than other models explored.



Confusion Matrix

	Predictions	
Actual	**0**	**1**
0	2710	162
1	298	1584



Cost Matrix

	Predictions	
Actual	**0**	**1**
0	$0	$1,620
1	$149,000	$0

Appendix

As with any technology; the system required to generate meaningful data, cleaning of the data, as well as the models that were implemented require various software packages. The raw “code” used to explore the methods detailed above can be found in the following libraries:

- GitHub Repository
- R Markdown Results
- R Markdown for Random Forest Only