HW #5 – Cell2Cell

Data

- "cell2cell.csv". The dataset is on eLearning.
- The data are documented in the "Cell2Cell Data Documentation.xls" spreadsheet (eLearning).
- The data consists of customers divided into calibration and validation sub-samples contained in the data file called "cell2cell.dta". The calibration data contain customers, approximately half of whom churn. The rest of them are in the validation data, with a small portion of them churn. The roughly 50% churn rate in the calibration data is not realistic, but the "oversampling" of churners makes it easier to identify the profile of churners as distinct from the profile of non-churners. The validation data contain a small percentage of churners, which is the current monthly churn rate at Cell2Cell.

General Instructions

- This case requires you to develop a model for predicting customer churn at "Cell2Cell," a fictitious wireless telecom company, and use insights from the model to develop an incentive plan for enticing would-be churners to remain with Cell2Cell.
- The data for the case are a scaled-down version of the full database generously donated by an anonymous wireless telephone company. There are over 65 potential predictors. Assume that logistic regression is the best choice to develop your predictive model.
- This case requires both statistical analysis and creativity/judgment. I recommend that you not spend too much time fine-tuning your predictive model; make sure you spend enough time interpreting and using the results.

Write-up Instructions

1. Run a Logistic Regression Predictive Model

a) Read in the "cell2cell.csv" file to R and answer the following questions:

- a. How many customers are in the data (i.e., the total number of rows)?
- b. How many are in the calibration set (calibrat == 1)? How many are in the validation set (calibrat == 0)?
- c. What is the exact churn rate (up to 4 decimal points) in the calibration set?
- d. What is the exact churn rate (up to 4 decimal points) in the validation set?

b) Data cleaning:

- a. Select relevant columns (churndep, revenue:retcall, calibrat). Here, a:b means consecutive variables between a and b columns. Note that it is "recall", not "retcalls".
- b. Split the data into 2 datasets: training dataset if calibrat == 1, and validation dataset if calibrat == 0. Drop calibrat columns in both datasets.

c) Run a logistic regression model with "churndep" as your dependent variable in the training data. All the variables below "churndep", i.e., from "revenue" to "retcall" can be used as independent variables (e.g., revenue:retcall). You can select all other variables. This can be easily done by using a period instead of typing each X variable:

Interpret 2 largest odds ratios and 2 smallest odds ratios. Make sure to check the corresponding p-values.

d) Use the model to predict attrition probabilities in the validation data and show the 5 highest attrition probabilities. (Hint: First, arrange the attrition probabilities in descending order).

2. Determine and Rank the Economic Importance of the Predictor Variables

a) Produce a data frame (i.e., data matrix) with odds ratios and p-values as 2 columns.

- a. First, convert parameter coefficients from the earlier logistic regression to odds ratios and save them in a variable (let's say 'OR').
- b. You can extract p-values with the following command. Note that there is no end bracket.

```
coef(summary(churnmodel))[,'Pr(>|z|)']
```

  Save them in a variable (let's say 'pvalues').

- c. Combine OR and pvalues into a data frame called df1 using `data.frame` command.
- d. View this data frame to make sure that it looks as expected.

b) Calculate standard deviations of the variables in the calibration data (e.g., you can use `sapply()` function) and save them as a data frame

   a. Check and see if any of the standard deviations are missing when the variable is not empty.

   b. If so, this may be because some of the values are missing. To resolve this issue, calculate standard deviations based on non-missing values using `function(x)` and `na.rm = TRUE` options.

   c. Again, use data.frame command to save this as a data frame called df2.

c) Merge df1 and df2

   a. First, add a column of row names (in our case, variable names) to both df1 and df2:
      `df1$VarName <- row.names(df1)`

   b. Use `merge` command (or something similar) to merge df1 and df2 based on matching "VarName". Keep only the matched ones using `all = FALSE` option.

d) Round the numeric columns of the merge data frame to 5 decimal points using `round()` command and keep rows with p-values < 0.05 only.

e) Export the results into a CSV file using `write.csv()` command.

f) Open the CSV file in Excel and determine the economic importance of each variable for predicting attrition (see lecture slides for the formula). Note the difference between a dummy variable and a non-dummy variable in the formula. See variable descriptions in the spreadsheet "Cell2Cell Data Documentation.xls". Be careful with variables that have similar names.

3. Create a Contingency Based Incentive Plan

a) Looking at the top 7 importance-ranked list of predictor variables, decide which of them suggests a retention action by Cell2Cell (i.e., actionable at the retention department).

b) For each actionable and statistically significant predictor variable, specify what retention action you suggest, i.e., what type of incentive you plan to give consumers to encourage them to remain with Cell2Cell (e.g., new phone, rebate, new plan). You do not need to provide the costs of the incentives. Just make sure that your incentives are reasonable and in accordance with "actionable" predictor variable.

c) For each non-actionable and statistically significant predictor variable, specify how to use the intelligence obtained (e.g., share the information with other departments, and if so, which department).