Day & Ross Freight: Sales Opportunity Win Prediction, an Analysis of Salesforce Data

Alexei Marcilio, Merrimack College

Mar 13th, 2018

Contents

Executive Summary

Salesforce.com opportunity data from the Canadian transportation and logistics company Day & Ross Freight Inc. was examined. Exploratory data analysis was completed and the machine learning techniques of logistic regression and random forest were applied in order to build a model predicting whether an opportunity would be won using various predictors such as opportunity age, calls, account calls and user tenure. Feature engineering was employed to create several of these independent variables. Half of winning deals (Closed/Won opportunities) were found to be closed in the first two months, and more deals were found to be won than lost in the first 70 days, the opposite being true thereafter. Call averages were not found to have a significant impact on win rates, and a large variation in win rates between departments was found. Logistic regression was used to predict a winning opportunity with 67% accuracy and random forest with 78% accuracy. The most important variables in building the random forest model were found to be the age of the deal, it's total projected revenue, the total calls a user made (in general, not related to the deal), and the age of the account of the deal. Data quality was found to be a significant issue, including the likelihood that many calls are not being recorded and win rates are exaggerated. The logistic regression and random forest models were applied against new data (deals that are not yet closed) and a likelihood of success and random forest prediction was determined for each of these deals. Recommendations were made on how to better leverage Salesforce.com and to utilize the results of these machine learning models.

Introduction

Headquartered in Hartland, New Brunswick, Day & Ross Freight Inc. is a Canadian transportation and logistics company that was founded more than 65 years ago. It has grown from a few trucks hauling goods in eastern Canada into one of the nation's largest national transportation providers. It is a wholly owned subsidiary of McCain Foods Limited (Day & Ross Transportation Group, 2017).

Companies that apply data driven decision making perform better than those in which decision makers rely on subjective models based on experience (Provost and Fawcett, 2013). Day & Ross Inc. have used Salesforce.com's Sales Cloud for several years, and through it, sales representatives and executives at the company can access various reports and dashboards summarizing data on the sales pipeline (Finelli, 2018). A move up the Analytics Maturity Model (TDWI, 2016) through the application of machine learning techniques could allow Day & Ross Inc. to better leverage their Salesforce.com data.

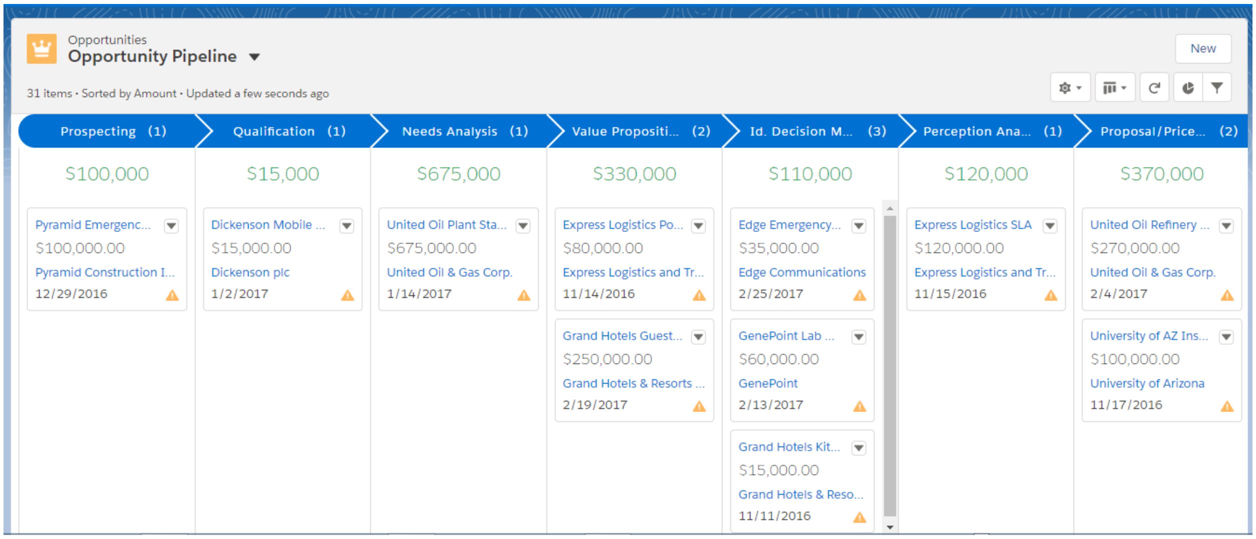
This report presents the results of the the machine learning models that were applied to predict whether an opportunity would be won or lost and the factors that have influence. Two models were built using logistic regression and random forest. In addition the data was explored, visualized, cleaned and appropriate features were selected.

Business Problem

A sales pipeline is a method of selling based on the fundamental principles of the sales process. It describes the sequence of steps that a sales person takes from first contact with a potential customer, to qualifying that prospect as a lead, to the validation of that lead into an opportunity, and moving through further stages until it is closed. Day & Ross monitors this sales pipeline using Salesforce.com, the world's largest CRM software vendor (Columbus, 2016). Figure 1 shows a Kanban Chart displayed in a typical Salesforce.com's Sales Cloud application showing the current pipeline of opportunities and their values.

The use of machine learning technology in business is predicted to double by the end of 2018 (Deloitte, 2017). The major CRM players are all betting that machine learning and artificial intelligence will be the next major industry disruptor, and have already implemented machine learning software in their CRM systems, including Oracle's Adaptive Intelligence, Microsoft's Cortana Intelligence Suite, and Salesforce's Enstein AI (Evans, 2018).

Day & Ross, much like any company, would like to increase the number of opportunities that they win. There are many demands on a sales representative's time and by scoring opportunities with the probability of success a rep can concentrate on those deals that are more likely to succeed and waste less time on those with very little chance of success. In addition, by knowing which factors are most correlated with winning an opportunity a business can focus on those activities that are most likely to move the needle.



Opportunities
Opportunity Pipeline ▼

New

31 items • Sorted by Amount • Updated a few seconds ago

Prospecting (1)	Qualification (1)	Needs Analysis (1)	Value Propositi... (2)	Id. Decision M... (3)	Perception Ana... (1)	Proposal/Price... (2)
$100,000	$15,000	$675,000	$330,000	$110,000	$120,000	$370,000
Pyramid Emergenc... $100,000.00 Pyramid Construction I... 12/29/2016	Dickenson Mobile ... $15,000.00 Dickenson plc 1/2/2017	United Oil Plant Sta... $675,000.00 United Oil & Gas Corp. 1/14/2017	Express Logistics Po... $80,000.00 Express Logistics and Tr... 11/14/2016	Edge Emergency... $35,000.00 Edge Communications 2/25/2017	Express Logistics SLA $120,000.00 Express Logistics and Tr... 11/15/2016	United Oil Refinery ... $270,000.00 United Oil & Gas Corp. 2/4/2017
			Grand Hotels Guest... $250,000.00 Grand Hotels & Resorts ... 2/19/2017	GenePoint Lab ... $60,000.00 GenePoint 2/13/2017		University of AZ Ins... $100,000.00 University of Arizona 11/17/2016
				Grand Hotels Kit... $15,000.00 Grand Hotels & Reso... 11/11/2016		

Figure 1: Kanban chart of open opportunities in Salesforce.com's Sales Cloud.

Data

Salesforce.com data is contained in standard and custom objects, which can be thought of in database terms as tables. All records for the objects relevant to this project were exported and provided to the author using Salesforce.com's Apex Data Loader (for events and tasks), or by means of a custom report.

Table 1 Shows the Salesforce.com objects used in this study. Opportunity is the main object of interest and contains fields such as the stage, the reason, the expected revenue and the start and close date of the opportunity. The other tables are related to the opportunity table through lookup fields (this is akin to the referential integrity of a database).

The user table, related through the User Id, shows details about the owner of the opportunity, and the account table, related through the Account Id, shows details about the associated account. The event and task tables are used interchangeably and display details about the activities a sales representative has with a client, for example, if they call or email them. These activity tables are related to the opportunity through the WhatId field. The contact table shows information about the people who work at the business that's related to the opportunity.

Table 1: All the records from the following Salesforce.com objects in Day & Ross's instance where provided for this report.

Salesforce Object	No. of Records	Description
Opportunity	16,913	All opportunities, their names, potential revenue, stage, start date and closed date, type, and reason.
Account	53,985	The account associated with the opportunity - each opportunity references only one account.
Contact	53,161	The contact from the client's company referenced by the opportunity.
User	180	The Day & Ross employee who owns the opportunity - each opportunity is owned by only one user.
Event	64,275	The activities related to an opportunity, or the associated account, for example a call or email.
Task	770,532	Tasks are used interchangeably with events so the tables were combined.

Features and Target

Implisit Insights, a data intelligence company bought by Salesforce.com in 2016 analyzed 21,000 opportunities across various industries and found several things that related to winning opportunities; these were involving more than one sales rep, time - the longer an opportunity was open the less likely it was to be won, engaging more people from the client side, and a greater amount of communication around the opportunity.

Day & Ross currently limits an opportunity to one sales rep, and does not regularly record client contacts associated with the opportunity, so some of these things will be difficult to test. The target variable will be the opportunity stage of closed/won (1) vs. those that are closed/lost(0).

After carefully considering the data quality of many of the features, and the fact that many categorical fields had too many values, Table 2 shows the final set of features used for this report.

Table 2: The features chosen for analysis and their descriptions.

Feature	Type	Table	Description
Type	categorical	Opportunity	New Business or Expanded
PotRevenue	numeric	Opportunity	Estimated Revenue
AccCalls	numeric	Event & Task	# Total Calls to Account
AgeInDays	numeric	Opportunity	Age of Deal (Days)
TotOpCalls	numeric	Event & Task	# of Opp. Calls
UserTotCalls	numeric	Event & Task	Lifetime # Calls by User
UserTenure	numeric	User	Tenure of Owner (Days)
AccountAge	numeric	Account	Age of Account(Days)
NumContacts	numeric	Account & Contact	# Total Contacts Called @ Account
NumOppEmails	numeric	Event & Task	# Opp Calls that were Emails
NumAccEmails	numeric	Event & Task	# Account Calls that were Emails
CanOrNot	categorical	Opportunity	Opp in Canada or Not

Data Preparation

The following steps were completed in order to create a suitable data set for analysis:

1. Each table provided as a .csv file was loaded into a MySQL database.

2. Queries was run to check for duplicate data. Although these tables are unique on the Salesforce Id fields when extracted in reports duplicates can occurs as lookup fields are resolved as addition records. This does not occur when extracted with the Apex Data Loader. No duplicate data was found.

3. Queries were used to find the number of unique values, inconsistent data, missing values and incorrectly entered data. For example the city field in accounts was found to have many data related issues. These descriptive fields which were entered freely by reps, such as description, subject, and city were not used in modeling.

4. Data sets were loaded into R-Studio for additional analysis.

5. Histograms were created of the numeric variables considered to examine the distribution of these values (Figure 2). These show some interesting patterns. All the activity data is skewed to the right, in other words where are many low values, even 0 recorded calls or email for the accounts associated with the deal. Total Opportunity calls, which are calls directly associated with an opportunity, are surprising low. It's obvious many calls are not being recorded. User Tenure and Account Age spike at certain ages and this could be a result of these records being added when Salesforce was initially launched. The user's average daily calls are also skewed, which indicates some reps are using salesforce properly and entering all their calls, and others are not.

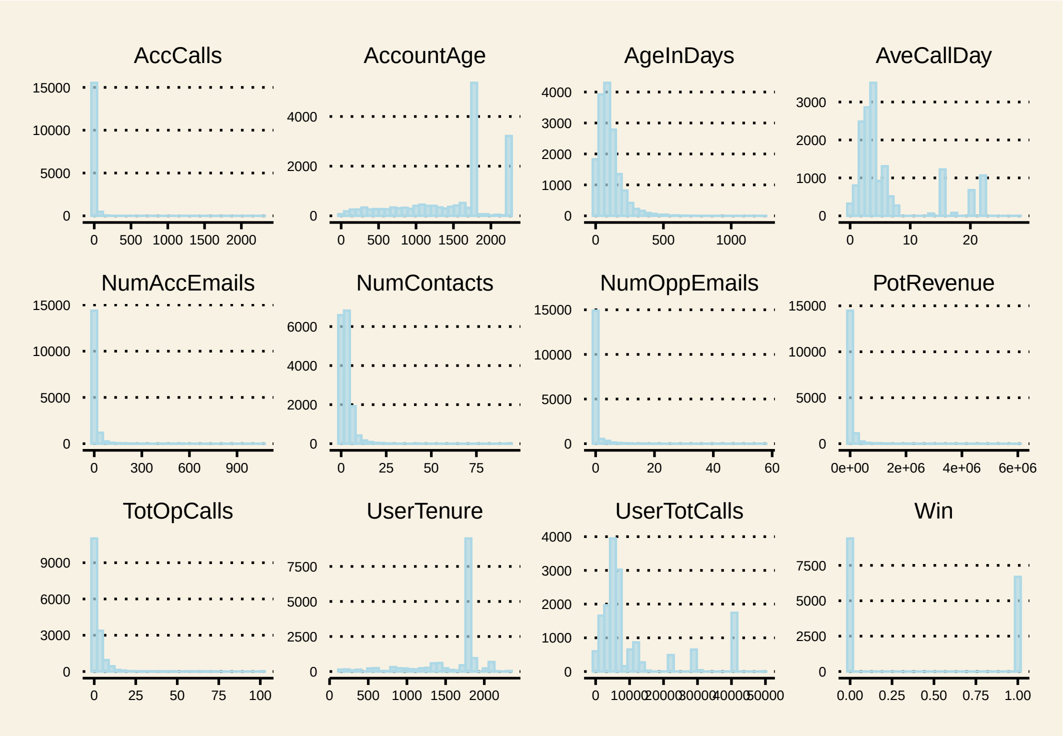


Figure 2: Histograms of the distributions of each numeric feature used in the final data set.

Let's examine the distributions of the various sales stages. Figure 3 shows each stage and the distribution of their ages. There's several things about these plots that are curious. First Prospecting is the earliest stage in the process, and it reflects the probability that we think an opportunity might exist but we need further qualification to make this determination. This phase should never be longer than a couple months, let alone some of the time periods we see here. Second, closed opportunities have the relatively short average ages, and there are many Closed Won opportunities that have an age in days of 0 indicating some reps are only entering deals when they know they've won them.

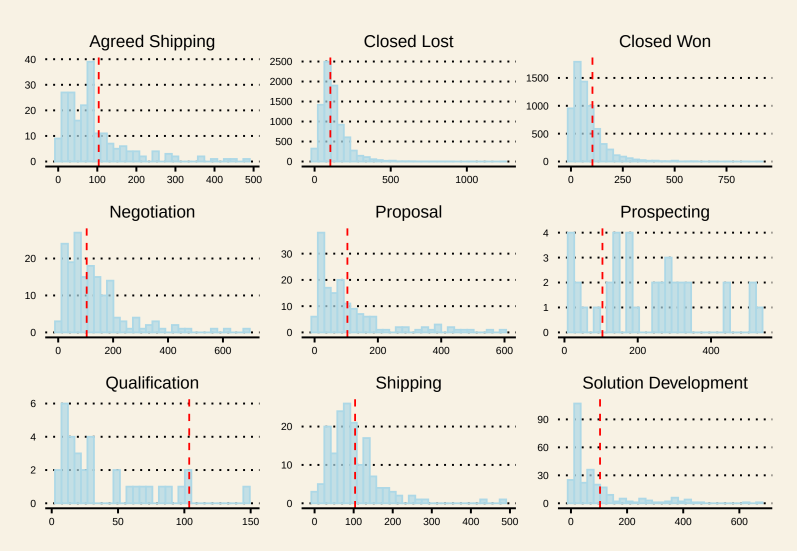


Figure 3: Histograms of the distributions of the age of each stage in the pipeline. Mean values shown by red line.

Data Cleaning

We used the Amelia package to quickly visualize if there are any missing values in our data. No missing data was found. The R code and output for this section and the visualizations are show in Appendix B.

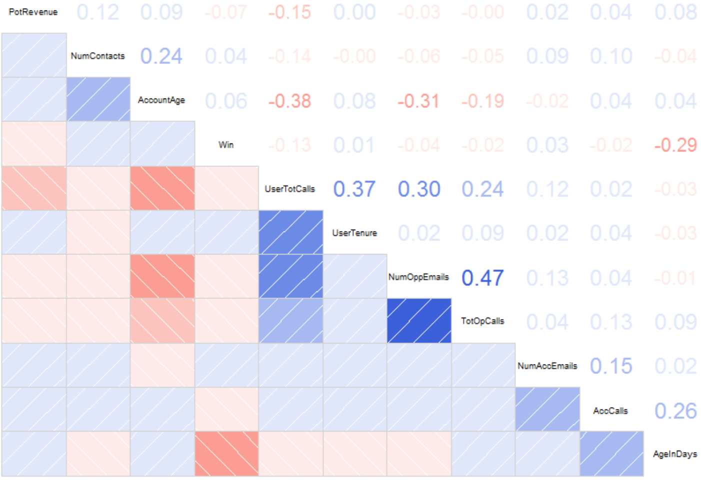


Figure 5: The corrgram shows the relationship between the predictors, and how correlated they are to each other.

Figure 5 shows that the predictors are not overly correlated, the highest r-squared value being 0.47. We can see that Opportunity Calls and Emails are moderately correlated, which certainly makes sense as the more often a client is called, the more frequent the number of emails exchanged. What does not make sense is the negative correlation between activities and winning a deal. The most logical explanation is that many reps are not entering their tasks and events, particularly those reps that are winning more deals. This is consistent with many winning deals are closed after only one day. It is likely that these reps are simply recording only those deals that they have already won. Gaming the system in this way undermines the benefit of Salesforce, reducing it's ROI significantly.

We can see the the age of the deal is negatively correlated with winning it. This is expected as many studies have shown that the older the deal the less likely it is to be won. It's also a result of the above mentioned data issues.

Results

Data Exploration

Let's visualize the data to gain a better understanding of deals and how they are won. Figure 6 shows that there's a clear difference between the age of deals which are won and those which are lost. The average age of wining deals is almost half that of the losing deals.

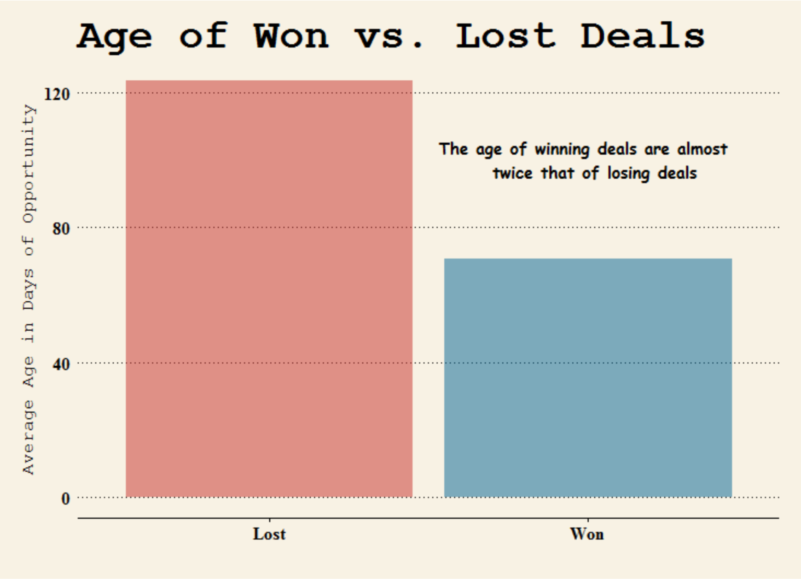


Figure 6: A comparision of the average age in days between deals that were won and deals that were lost.

Figure 7 shows that most won deals (green line) are won within 2 months, and lost deals (red line) can linger. There's a spike in deals closed at around the 6 month period, perhaps due to an assessment of all deals at that time, or a natural deal cycle. It's unusual not to see a curve in the winning line share, as the largest share of deals are closed within a couple days. This indicates that the reps are not entering the deal data correctly, they are skipping stages, closing deals immediately after entering them, and not consistently entering losing deals.

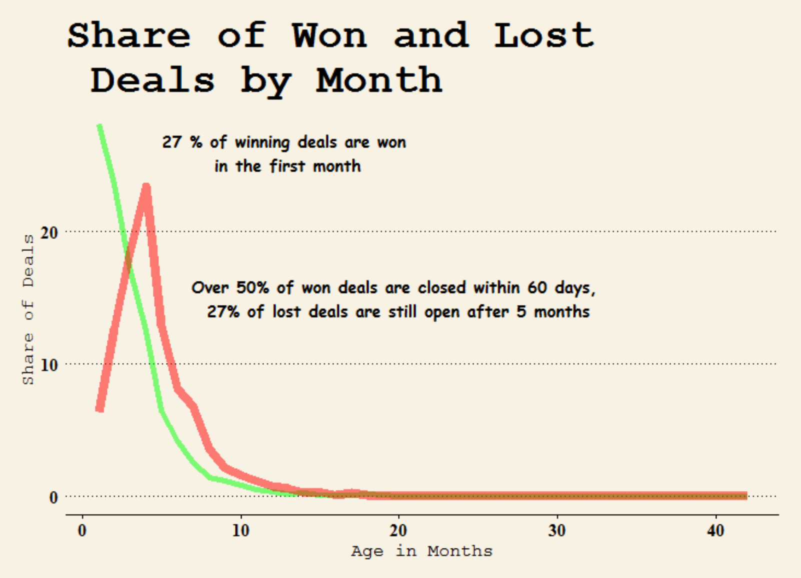


Figure 7: The share of all deals won by age of the deal in months. Green line is the share of won deals, and Red line shows the share of lost deals.

Figure 8 shows that the likelihood of a deal closing diminishes with time. After 70 days chances are the deal won't be won.

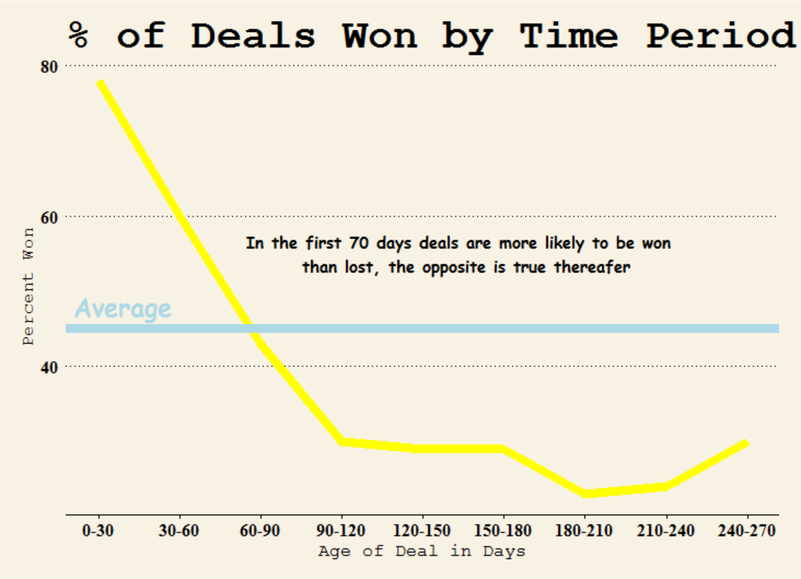


Figure 8: The percent of deals won by their age in days (yellow line). The average percent won is show with the blue line.

If we compare various features between the deals that were won and those that were lost we see very little difference. Figure 9 shows that there's no significant difference of those measures between winning and losing deals. Calls are skewed to the right, meaning there are a great deal of opportunities with very few or no calls made where the opportunity is associated with the task or event. It's likely that many reps are not properly indicating which opportunity their call is associated with. This makes it difficult or impossible to assess and analyze the value of the interaction to whether a deal was won. The calls may be of course made instead against the account, however that account lives beyond the opportunity.

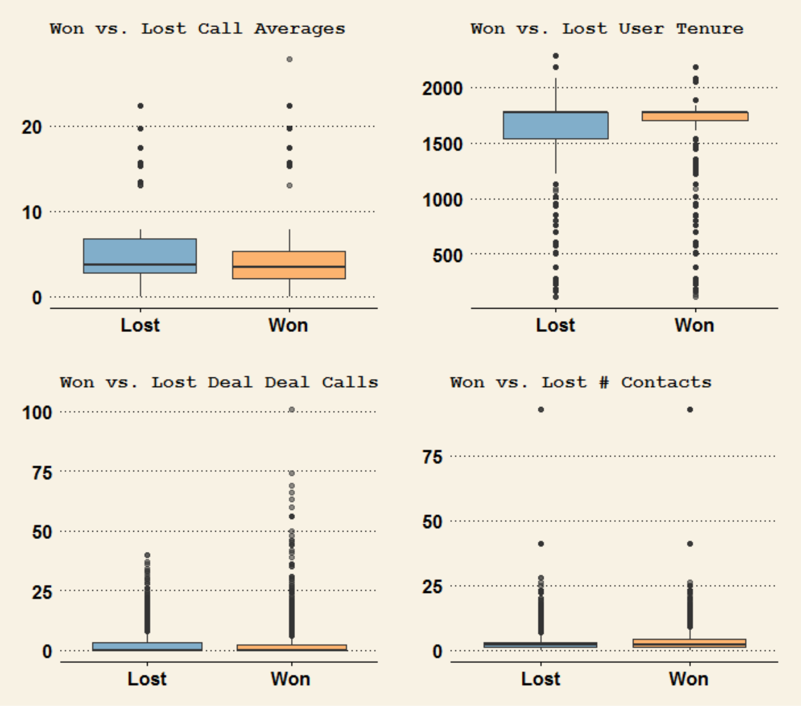


Figure 9: Boxplots of Call Averages, User Tenure, Deal Calls, and # Contacts between winning and losing deals.

Figure 10 shows that win rates vary significantly by department from over 60% for the Quebec Freight Group to under 30% for the USA Freight group. In many industries a win rate of over 30% suggests that only those deals that are already advanced in the selling cycle are being entered into Salesforce.com. Win rates this high are suspicious.

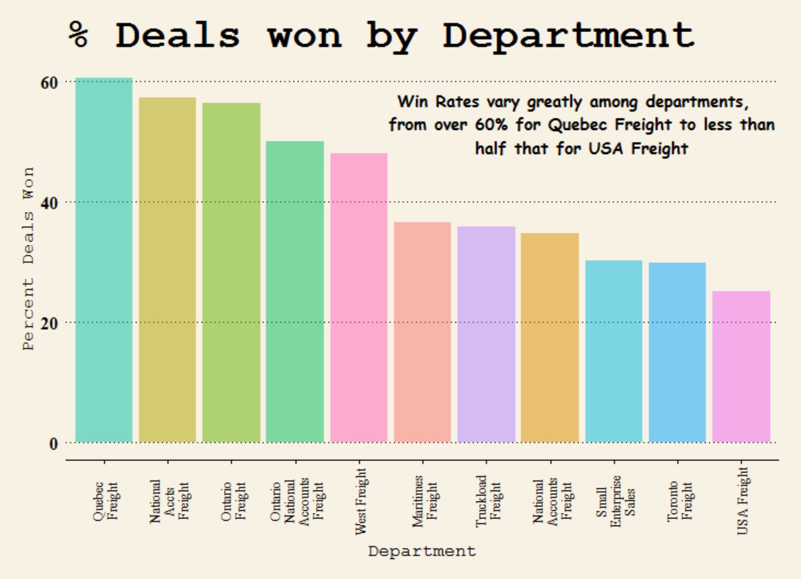


Figure 10: The difference in the percent of deals won by department. Win rates vary significantly from over 60% to under 30%.

Figure 11 shows that departments that are closing the big deals do so fairly consistently each quarter. Although this shows potential revenue of the deal, we could assume there's a high correlation between this and the eventual actual revenue realized.

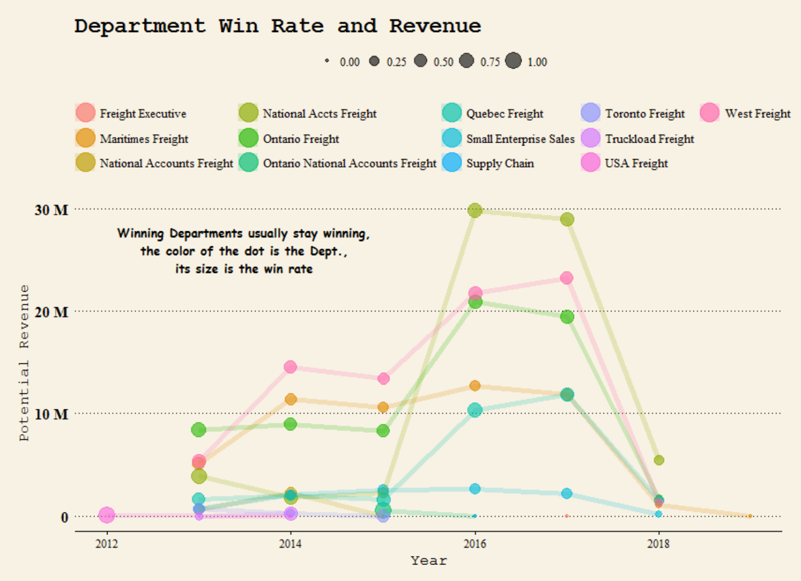


Figure 11: Win Rates by Department by Year. The size of the dot represents the win rate. There's consistency amond winning departments each year.

By the same token winning reps are fairly consistent also. Figure 12 shows the top 6 reps by winning revenue and how they perform each year. Except for a pickup by National Accounts Freight, it's interesting that the rank is almost fully preserved each year, as we can see by the fact the lines rarely overlap.

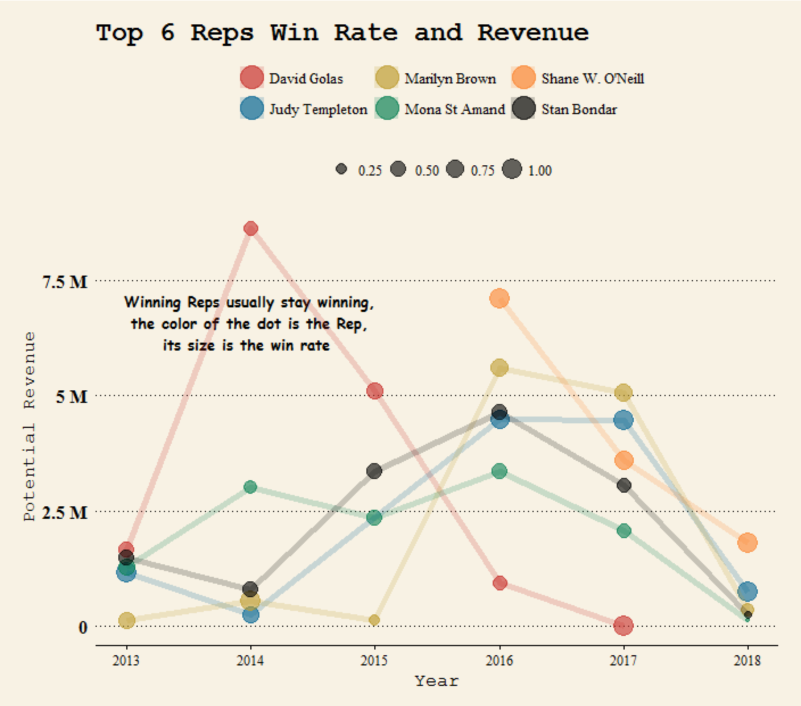


Figure 12: Top Reps potential revenue won by year. The size of the dot represents the win rate.

Machine Learning Models

In the last several years machine learning has developed into a powerful tool with applications in many businesses. It is now recognized in fields from healthcare, to genetics, social sciences and business. Machine learning models will now be applied to the Day & Ross data to predict winning deals. We can think of a model as any function that has predictive power (Burger, 2018). We will divide the data we have into a training data set and a test data set. Machine Learning compels us to train a data model, in other words build the model using the training data set, and then test this model on unseen data, the test data set.

The R code and output for this section is show in Appendix A.

Data Preparation

Let's look at the structure of the data. We can see that the predictors have the correct data types. Those that should be factors, and those that should be numeric are correctly designated. Stage and Type and CanOrNot(whether a deal is Canadian) are factors, and the rest of the predictors are numeric.

We first must filter the data to only include the closed opportunities. Then we will remove the Stage feature, which was only used to subset the data with targets used for training and test.

Now we split the data into a training set and a test data set. We will first make the target a factor. We then split the data into 70% train and 30% test. This results in 11079 records in the training set and 4748 records in the test data set. The training data set is used to build the model and the test set will then be used to test the accuracy of our predictions against known results (whether or not the deal was actually won).

Logistic Regression

Now let's build the model on the training data set. Logistic regression is a machine learning method of analyzing a dataset of one of more independent predictors that determine an outcome of a target which is a dichotomous variable (in this case the two possible outcomes are Closed/Win and Closed/Lose) (James et al, 2017). In other words we want to predict whether a deal is won using the other fields that we have either summarized or derived.

Using the summary function revealed that ten of the coefficients were significant, which makes the model a little complex. It makes sense that the age of the opportunity is negatively correlated with winning, however the other coefficients are surprising. There's a negative correlation with potential revenue, total user calls, total opportunity emails and account emails, which could related to data quality issues. It's not surprising that total opportunity calls is positively correlated with winning a deal.

Now let's use the model to predict the winning deals in the test set. We can compare the predictions with the actual results to find out how accurate the model is. We first use the predict function against the test data set. This produces probabilities for each deal. We then convert them to 1 or 0 - if a deal has a probability greater than 0.5 we convert it to a 1, the rest are converted to 0.

We now can determine the misclassification error of the model. This is the accuracy of our predictions. The result it 67%. These results are fairly good. The fact that it was not higher is likely a result of poor data quality which will be discussed later in the paper. The confusion matrix is then produced (Table 3). This will show us the number of false positive and false negatives in our predictions.

Table 3: The confusion matrix of the logistic regression model test set predictions.

	False	True
0	1710	745
1	809	1484

In this case the positive class is 0. The positive predictive value is calculated as the number of correct positive predictions out of the total number of positive predictions (Wang, 2013). In this case it's 71% slightly higher than the negative predictive value of 65%. Therefore using our model is more useful at determining a losing deal than at predicting a winning deal. With this in mind our model could be considered more useful at screening for losing deals, which might help a rep spend less time on deals that ultimately will not be won.

Using the ROCR library we can now show a receiver operating characteristic curve (ROC). It's a plot that illustrates the ability of a binary classifier as it's discrimination threshold is varied (James et al, 2017). It's a plot of a true positive rate against a false positive rate at various threshold settings. The plot for our test data set predictions are show in Figure 13.

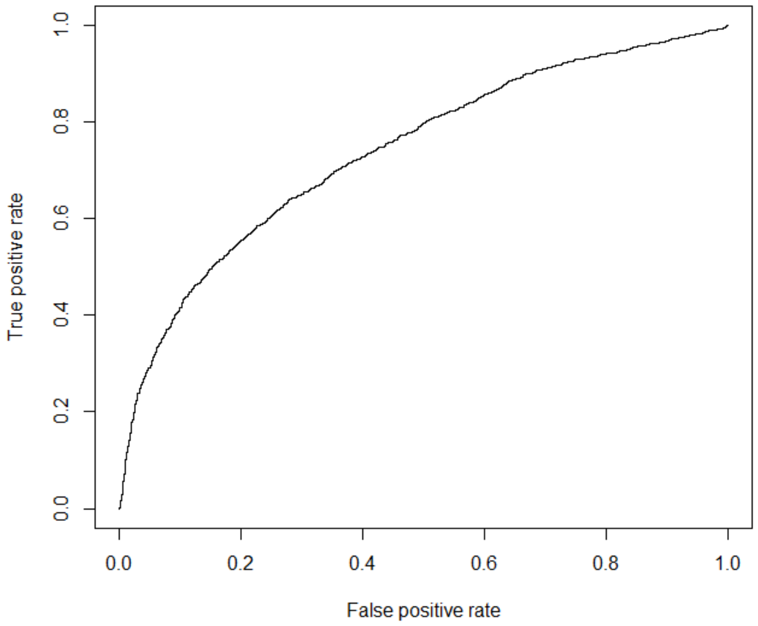


Figure 13: Receiver Operating Curve (ROC) for Lasso model of opportunity test set predictions.

We can check the area under the curve (AUC) using the performance function. This shows the overall performance of the classifier (James et al, 2017). In this case its 74%.

We can use the Lasso technique to reduce the number of variables. The Lasso technique is a fairly recent alternative to using ridge regression that overcomes the disadvantage of the former technique in that it always generates a model using all of the predictors (Chatterjee & Hadi, 2012). We can perform a grid search to find optimal value of lambda using the values of family=binomial and alpha=1.

The plot tells us that about 7 coefficients is optimal (Figure 14). This graph uses the cross validation version of the glmnet function in R, and it results in a plot showing the MSE (mean squared error) of several iterations of the model under changing values of lambda (the "regularization parameter") (James et al, 2017).

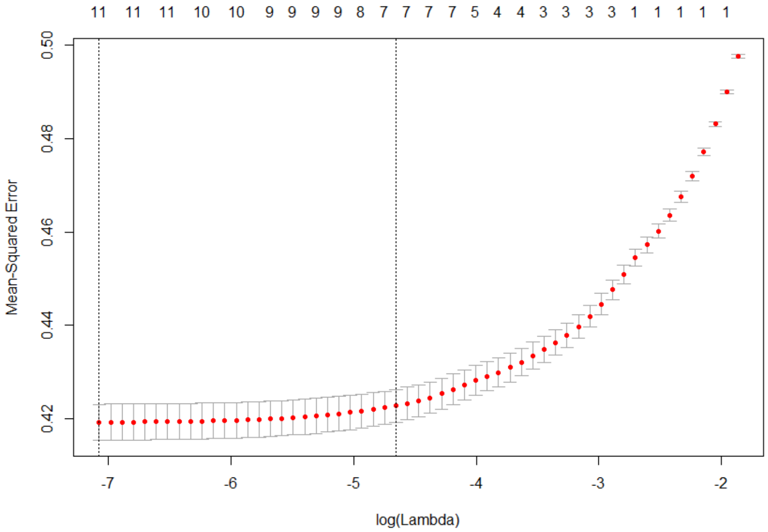


Figure 14: Graph of the mean squared error (MSE) of various iterations of the model given varying values of lambda (the regularization parameter).

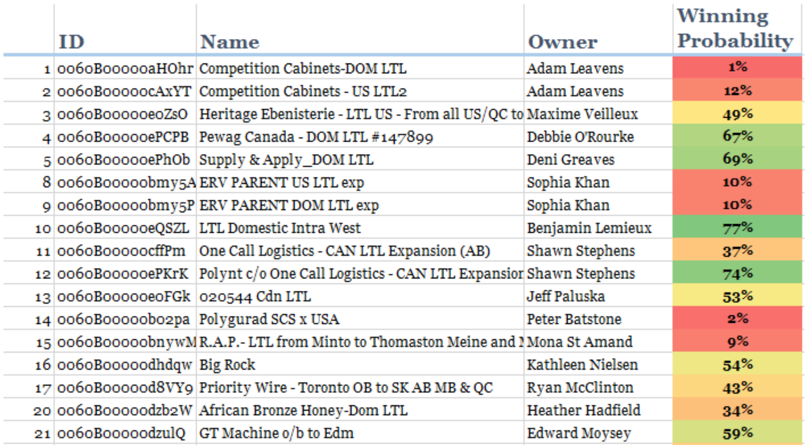
Now we can examine the coefficients. We have reduced the model to only 7 coefficients now. Let's test the accuracy of this model. The accuracy is about 67% which is about the same as we got before. The bias-variance trade off tells us that the simpler function should be preferred because it is less likely to over fit the training data.

New Data We can now use the model to predict whether open opportunities will be won or not. This time we should include the names and IDs of the opportunities so we can match them and make the information more useful to the sales rep.

Now we can use the predict function to predict whether these deals will be won. We will keep the probabilities and present this so a determination can be made as to which deals are more likely to be won. We have to exclude the three new columns as the model was not built using them.

Exporting this to excel makes these results more presentable. The probabilities could easily be loaded into Salesforce.com matching on the opportunity IDs. In this way reps could have immediate access to this information.

Part of the final file, conditionally formatted, is shown in Figure 15:



	ID	Name	Owner	Winning Probability
1	0060B00000aHOhr	Competition Cabinets-DOM LTL	Adam Leavens	1%
2	0060B00000cAxYT	Competition Cabinets - US LTL2	Adam Leavens	12%
3	0060B00000eoZsO	Heritage Ebenisterie - LTL US - From all US/QC to	Maxime Veilleux	49%
4	0060B00000ePCPB	Pewag Canada - DOM LTL #147899	Debbie O'Rourke	67%
5	0060B00000ePhOb	Supply & Apply_DOM LTL	Deni Greaves	69%
8	0060B00000bmy5A	ERV PARENT US LTL exp	Sophia Khan	10%
9	0060B00000bmy5P	ERV PARENT DOM LTL exp	Sophia Khan	10%
10	0060B00000eQSZL	LTL Domestic Intra West	Benjamin Lemieux	77%
11	0060B00000cffPm	One Call Logistics - CAN LTL Expansion (AB)	Shawn Stephens	37%
12	0060B00000ePKrK	Polynt c/o One Call Logistics - CAN LTL Expansior	Shawn Stephens	74%
13	0060B00000eoFGk	020544 Cdn LTL	Jeff Paluska	53%
14	0060B00000bo2pa	Polygurad SCS x USA	Peter Batstone	2%
15	0060B00000bnywM	R.A.P.- LTL from Minto to Thomaston Meine and	Mona St Amand	9%
16	0060B00000dhdqw	Big Rock	Kathleen Nielsen	54%
17	0060B00000d8VY9	Priority Wire - Toronto OB to SK AB MB & QC	Ryan McClinton	43%
20	0060B00000dzb2W	African Bronze Honey-Dom LTL	Heather Hadfield	34%
21	0060B00000dzulQ	GT Machine o/b to Edm	Edward Moysey	59%

Figure 15: Excel file of Deals with their predicted probability of winning conditionally formatted.

Random Forest

The next modeling technique applied is random forest. The random forest (Breiman, 2001) is an ensemble method that one can think of as a type of nearest neighbor predictor. Ensembles are a approach that divides and conquers to improve performance. The main point of ensemble methods is to allow "weak learners" to come together to form a "strong learner".

When we apply the random forest model and use it to predict against the test data set we can see that the accuracy is 77.5% which is quite a bit better than the logistic regression model. The confusion matrix is also shown (Table 4). As opposed to the other models this model has almost equally positive and negative predictive values (77.95% vs 77.72%) which means it's equally good at predicting winning as well as losing deals.

Table 4: The confusion matrix of the random forest model test set predictions.

	False	True
0	2069	614
1	450	1615

Most studies on random forests are difficult to interpret since they are typically treated like a black box (Palczewska, 2014). It's true that a forest is made up of many of deep trees, and each of these trees is trained using a random selection of features, so obtaining a complete understanding of process by which decisions are made by examining each individual tree is impractical. A tree with a depth of 500, as in this case, can have tens of thousands of nodes, which means it's impossible to use as an explanatory model.

We can however check the relative importance of each variable using the importance function in the random forest package (Figure 17). Here we can see that Age in Days (the age of the opportunity) is the most important feature, which makes sense given our previous findings. Potential revenue, account age and account calls are also important, however we do not know if they have a positive or negative effect.

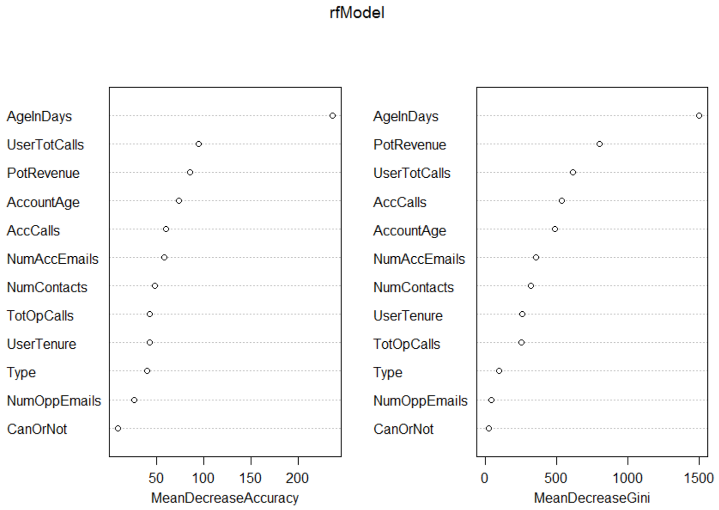


Figure 17: The relative importance of each variable using the random forest importance function.

New Data The random forest model had a prediction accuracy of almost 10% greater than that of logistic regression. It's worthwhile therefore to use this model to predict results based on new data, those deals which are still open. We can then combine these predictions with those obtained by the logistic regression model to provide more insight to Day & Ross.

The results are added to the data frame and exported to a csv file again (Figure 18) . Now we have a file with both the logistic regression and random forest results.

With the added prediction of the random forest model, the sales rep can better focus on those deals that have both a high probability score and a positive predicted outcome by the random forest model.

Figure 18: Excel file of Deals with their predicted probability of winning and the random forest model predictions conditionally formatted.

Validity & Reliability Assessment

We have examined the accuracy of the logistic regression and random forest models against the test data sets and found that they had a prediction accuracy of 67% and 78% respectively. Another method to evalute the model is to test how well it performs on certain subsets of the data. A popular way to do this is k-fold cross-validation in which the data is partitioned into k segments (or folds) of equal size (Burger, 2018). We hold out one segment for validation and use the other k-1 segments to train the data and we then repeat this k number of times. We track the accuracy of each model in predicting the set of data that is held out.

When we perform the k-folds validation the model shows a summary of different samples sizes, 9972, 9972, 9971, 9971, and 9970 etc. The accuracy of the model is 67.7% which is roughly what we obtained previously. This gives us a greater amount of confidence in the model. The confusion matrix of this k-folds cross validation model shows higher performance predicting losing deals (70%) than winning deals (65%), much like our previous model.

Recommendations

Based on the analysis and modeling of the data I can make the following recommendations in order to better leverage Salesforce.com in the future.

Machine Learning

Machine Learning and AI are growing at a tremendous pace. International Data Corporation (IDC) predicts that spending on AI and ML will grow from $12B in 2017 to 57.6B by 2021 (IDC, 2016). Salesforce.com has described how we are currently living in the fourth industrial revolution on business in which data is the new currency and artificial intelligence is driving innovation where the more data you have the better your predictions (Salesforce.com, 2018).

Salesforce.com's Einstein Analytics is an option for many companies with Salesforce to leverage AI however it's expensive and best used when there's universal buy in for Salesforce.com and data quality is high. That's not the case at Day & Ross.

I recommend the following actions:

1. Use the results of the logistic regression and random forest predictions for open deals by creating a custom field in Salesforce.com and apply the same conditional formatting as in the excel file provided. Load the results into Salesforce.com by matching the opportunity IDs[1]. In this way sales reps can see which deals are most likely to be won and concentrate their efforts on these. A sales rep's time is limited and to be most productive it's usually best to focus on deals with the highest likelihood of success and ignore those with the lowest.

2. Revisit machine learning to predict success for deals in the future once data quality issues are addressed. Despite the data quality issues the random forest model predicted the outcome of a deal with 78% accuracy, which is not bad considering many calls were not properly captured and sales reps skipped phases, likely only entering deals once they had already progressed through the pipeline, and closing a great many deals in 0 days. Once these issues are resolved it's likely that the predictive accuracy of the model will increase and the factors which influence whether a deal is won will be better understood.

3. Leverage machine learning and analytics on other parts of the business. For example lead scoring can be done in a similar way. A prediction of whether a lead will be converted might be useful to know which leads warrant greater effort, and what it is that made for a successful lead.

4. Include more sales reps in deals. At Day & Ross Freight there's only one owner of each opportunity, however it's often wise to include different employees at various stages of the deal depending on their strengths. This should be recorded so it can be analyzed in the future. It's been shown that involving more team members in a deal correlates with success.

5. Capture the actual revenue from a deal, not just the potential revenue as estimates can be inaccurate. Having the actual revenue would allow for more meaningful analysis and using the actual revenue as a target for a machine learning algorithm would be a very interesting future project.

Data Quality

Data quality is obviously something that must be addressed by Day & Ross. The following data quality related issues were observed and recommendations on how to resolve these issues are given:

[1] The opportunity ID found in reports and through the Salesforce interface is 15 characters, when Apex Data Loader is used to extract opportunities the opportunity ID will be 18 characters as it includes a trailing 3 character checksum. This can be removed in order to match IDs.

1. All opportunity calls were not recorded.

   There are relatively very few calls made that were directly associated with the deal. It's important to record this so that the benefit of calls can be assessed in the future and so that sales reps have a record of these calls in order to help them progress in closing deals. The fact that opportunity calls were not found to be well correlated with success is very surprising, and it does not mean that the fewer calls the better, instead this indicates that calls were not being properly recorded. This is reinforced by the finding that the overall number of calls that a sales rep makes is in fact an important factor to the random forest model.

2. Data quality overall is very poor.

   City fields have many misspellings, postal codes are missing or inaccurate, start dates of many opportunities are later than the end dates, many fields are missing when they should be required, such as the subject field in the task and event objects. In order to alleviate these issues I recommend:

   (a) Converting some free form text fields to list boxes. Although analysis can be done on text fields it's more straightforward analyzing fields that have limited options. In addition this will help ensure privacy rules are adhered to as free form text fields can be a privacy risk.

   (b) Making several fields such as subject in tasks and events required. As the fields currently have missing the first step would probably be to update the fields with values using the apex data loader prior to making the field required.

   (c) Adding validation to many fields. This will ensure end dates come after start dates, that an opportunity for each activity is chosen, and that postal codes of addresses are correct when entered.

   (d) The task and event objects should be consolidated. There seems to be no reason to use both. All events should be loaded into task and task should be used exclusively in the future.

   (e) The Prior Stage of the pipeline should be captured in Salesforce with a custom field and dashboards to obtain full visibility of the pipeline should be created. A company should encourage trust so there's no fear at all in setting a deal to Closed Lost. In many types of businesses having a win rate over 30% suggests that only sales deals that are already well established in the sales cycle are entered into the pipeline. This means a company misses out on pipeline visibility. This can be addressed by using dashboard charts to measure sales pipeline quality and by capturing the prior stage of a deal to ensure reps are moving through the stages properly.

Ultimately the most important thing by far to achieving a return on investment from Salesforce.com is executive buy in. Unless Salesforce.com becomes the system of record and there's incentive for using the system it will be difficult to realize value from the system. If a rep is successful management may believe the adage "if it's not broke why fix it". However the true value of CRM is discovering what makes a good sales rep good and imparting this knowledge to under performing reps. The only way to do this is everyone uses the system properly, entering all their calls. When this is done the value of Salesforce.com can be enormous.

References

Breiman, L. Machine Learning (2001) 45: 5. https://doi.org/10.1023/A:1010933404324

Burger, S. V. (2018). Introduction to Machine Learning With R Rigorous Mathematical Analysis. Oreilly & Associates Inc.

Chatterjee, S., & Hadi, A. S. (2012). Regression analysis by example. Hoboken, NJ: Wiley.

Columbus, L. (2016, May 28). 2015 Gartner CRM Market Share Analysis Shows Salesforce In The Lead, Growing Faster Than Market. Retrieved February 10, 2018, from https://www.forbes.com/sites/louiscolumbus/2016/05/28/2015-gartner-crm-market-share-analysis-shows-salesforce-in-the-lead-growing-faster-than-market/#7bc20aca1051

Day & Ross Transportation Group., 2017. Retrieved February 04, 2018, from http://www.dayrossgroup.com/

Deloitte Global 2017 TMT Predictions: Mobile Machine Learning expected to expand, helping to transform society | Deloitte China | Press release. (2017, January 22). Retrieved February 10, 2018, from https://www2.deloitte.com/cn/en/pages/about-deloitte/articles/pr-deloitte-tmt-2017-predictions.html

Evans, B. (2018, January 30). Oracle Places Huge Bets On AI And Machine Learning To Overtake Salesforce In SaaS. Retrieved February 10, 2018, from https://www.forbes.com/sites/bobevans1/2018/01/30/oracle-places-huge-bets-on-a-i-and-machine-learning-to-overtake-salesforce-com-in-saas/#19d037c81da1

Finelli, R., (2018, Feb 2). Personal interview. IDC, (2018). "Worldwide spending on cognitive systems forecast to soar to more than $31 billion in 2019, according to a new IDC spending guide," press release, March 8, 2016, www.idc.com/getdoc.jsp?containerId=prUS41072216.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning: with applications in R. New York: Springer.

Palczewska, A., Palczewski, J., Robinson, R. M., & Neagu, D. (2014). Interpreting Random Forest Classification Models Using a Feature Contribution Method. Integration of Reusable Systems Advances in Intelligent Systems and Computing, 193-218. doi:10.1007/978-3-319-04717-1_9

Provost, F., Fawcett, T., (2013). Data science and its relationship to big data and data-driven decision making. Big Data 1 (1), 51-59.

Salesforce.com., (2018). Impacts of the Fourth Industrial Revolution. Retrieved March 11, 2018, from https://trailhead.salesforce.com/modules/impacts-of-the-fourth-industrial-revolution/units/understand-the-impact-of-the-fourth-industrial-revolution-on-business

TDWI. (2016). TDWI Analytics Maturity Model and Assessment Tool. Retrieved February 11, 2018, from https://tdwi.org/pages/maturity-model/analytics-maturity-model-assessment-tool.aspx

Wang H., Zheng H. (2013) Positive Predictive Value. In: Dubitzky W., Wolkenhauer O., Cho KH., Yokota H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY

Appendix A

Machine Learning Models

Data Preparation

```
#Structure of the Data
str(Deals)
```

```
## 'data.frame':    16867 obs. of  14 variables:
##  $ Stage       : Factor w/ 9 levels "Agreed Shipping",..: 2 2 3 2 3 3 3 2 2 2 ...
##  $ Type        : Factor w/ 2 levels "Expanded","New Business": 2 2 2 1 2 2 1 2 1 2 ...
##  $ PotRevenue  : int  4800 24000 2400 20000 20000 25000 20000 20000 144000 60000 ...
##  $ AccCalls    : int  2 1 7 17 12 0 13 22 0 0 ...
##  $ AgeInDays   : num  101 64 52 45 37 13 58 83 221 98 ...
##  $ TotOpCalls  : num  7 2 3 0 3 0 6 0 0 0 ...
##  $ UserTotCalls: int  41063 1907 28836 12081 12081 12081 12081 12081 841 841 ...
##  $ UserTenure  : int  2079 950 1890 1771 1771 1771 1771 1771 1771 1771 ...
##  $ AccountAge  : int  534 534 534 534 534 534 534 534 534 534 ...
##  $ NumContacts : num  1 1 2 8 8 8 8 8 5 5 ...
##  $ NumOppEmails: num  5 0 1 0 0 0 0 0 0 0 ...
##  $ NumAccEmails: num  11 0 6 9 9 9 9 9 0 0 ...
##  $ CanOrNot    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Win         : num  0 0 1 0 1 1 1 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int  1079 1486 1775 2521 2523 4365 4493 4970 5226 5564 ...
##   ..- attr(*, "names")= chr  "1079" "1486" "1775" "2521" ...
```

```
# We first must filter the data to only include the closed opportunities.
DealsClosed <- filter(Deals,
                      (Stage == 'Closed Lost') | (Stage == 'Closed Won'), (Win == 0 | Win == 1 ))
DealsClosed$Stage <- NULL
# Remove NA Records
DealsClosed <- na.omit(DealsClosed)
```

```
# Train Test Split
set.seed(110)

DealsClosed$Win <- factor(DealsClosed$Win)

split <- sample.split(DealsClosed$Win,SplitRatio = 0.70)
Deal.train <- subset(DealsClosed, split == TRUE)
Deal.test <- subset(DealsClosed, split == FALSE)

# Numbers of rows of training and test
nrow(Deal.train)
```

```
## [1] 11079
```

```
nrow(Deal.test)
```

```
## [1] 4748
```

Logistic Regression

```
# Build the logistic Regression Model
DealLRmodel <- glm(Win ~ ., family = binomial(link='logit'), data=Deal.train)
```

```r
# Use the summary function to check the coefficients of the model
summary(DealLRmodel)
```

```
##
## Call:
## glm(formula = Win ~ ., family = binomial(link = "logit"), data = Deal.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6917  -1.0360  -0.4937   0.9985   3.9440
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.623e+00  1.669e-01   9.726  < 2e-16 ***
## TypeNew Business -3.462e-01  4.875e-02  -7.100 1.24e-12 ***
## PotRevenue       -1.922e-06  2.258e-07  -8.512  < 2e-16 ***
## AccCalls          3.998e-03  6.705e-04   5.962 2.49e-09 ***
## AgeInDays        -1.035e-02  3.419e-04 -30.265  < 2e-16 ***
## TotOpCalls        2.892e-02  6.271e-03   4.612 4.00e-06 ***
## UserTotCalls     -3.265e-05  2.294e-06 -14.237  < 2e-16 ***
## UserTenure        9.646e-05  6.014e-05   1.604  0.10874
## AccountAge       -1.213e-04  4.325e-05  -2.805  0.00503 **
## NumContacts       9.844e-04  5.819e-03   0.169  0.86566
## NumOppEmails     -8.113e-03  1.611e-02  -0.504  0.61449
## NumAccEmails      3.341e-03  7.098e-04   4.706 2.52e-06 ***
## CanOrNot1        -2.590e-01  1.177e-01  -2.200  0.02781 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15318  on 11078  degrees of freedom
## Residual deviance: 13555  on 11066  degrees of freedom
## AIC: 13581
##
## Number of Fisher Scoring iterations: 5
```

```r
# let's predict the winning deals using this model

fit.prob <- predict(DealLRmodel, Deal.test, type = 'response')
fit.res2 <- ifelse(fit.prob > 0.5,1,0)

misClassErr <- mean(fit.res2 != Deal.test$Win)
```

```r
# Determine the misclassification error of the model. This is the accuracy of our predictions.
print(1 - misClassErr)
```

```
## [1] 0.6727043
```

```r
table(Deal.test$Win, fit.prob > 0.5)
```

```
##
##     FALSE TRUE
##   0  1710  809
##   1   745 1484
```

```r
#confusionMatrix(fit.res2, Deal.test$Win)
```

```r
# Using the ROCR library we can now show a receiver operating characteristic curve (ROC).
library(ROCR)
p <- predict(DealLRmodel, Deal.test, type="response")
pr <- prediction(p, Deal.test$Win)

prf <- performance(pr, measure = "tpr", x.measure = "fpr")
# plot(prf)
```

We can check the area under the curve (AUC) using the performance function. This shows the overall performance of the classifier (James et al, 2017). In this case its 74%.

```r
# We can check the area under the curve (AUC) using the performance function
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```r
# Lasso technique to reduce the number of variables.
library(glmnet)

x <- model.matrix(Win~.,Deal.train)
y <- Deal.train$Win

cv.out <- cv.glmnet(x,y,alpha=1,family="binomial",type.measure = "mse")
#plot result
# plot(cv.out)
```

```r
#Now we can examine the coefficients.
#min value of lambda
lambda_min <- cv.out$lambda.min
#best value of lambda
lambda_1se <- cv.out$lambda.1se
#regression coefficients
coef(cv.out,s=lambda_1se)
```

```r
# We have reduced the model to only 7 coefficients now. Let's test the accuracy of this model.
#get test data
x_test <- model.matrix(Win~.,Deal.test)
#predict class, type="class"
lasso_prob <- predict(cv.out,newx = x_test,s=lambda_1se,type="response")
#translate probabilities to predictions
lasso_predict <- rep(0,nrow(Deal.test))
lasso_predict[lasso_prob>.5] <- 1
#confusion matrix
table(pred=lasso_predict,true=Deal.test$Win)
#accuracy
mean(lasso_predict==Deal.test$Win)
```

```r

# Let's use the model to predict whether open opportunities will be won or not.
dr.opportunities$Opportunity.Name <- as.character(dr.opportunities$Opportunity.Name)
dr.opportunities$Opportunity.Owner <- as.character(dr.opportunities$Opportunity.Owner)

DealsOpen <- select(dr.opportunities, Opportunity.ID, Opportunity.Name, Opportunity.Owner,
                    Stage, Type, Revenue.Potential, AccCalls, AgeInDays, TotOpCalls,
                UserTotalCalls, UserTenure, AccountAge, NumContacts, NumOppEmails,
                NumAccountEmails,Billing.Country, Win)

colnames(DealsOpen) <- c("ID", "Name", "Owner","Stage", "Type", "PotRevenue", "AccCalls",
                         "AgeInDays", "TotOpCalls", "UserTotCalls", "UserTenure", "AccountAge",
                         "NumContacts", "NumOppEmails", "NumAccEmails", "CanOrNot", "Win")


DealsOpen <- filter(DealsOpen, (Stage != 'Closed Lost') &
                      (Stage != 'Closed Won'), (Win == 0 | Win == 1 ))
DealsOpen$Stage <- NULL
DealsOpen <- na.omit(DealsOpen)
```

```
# Now we can use the predict function to predict whether these deals will be won.
fitOpen.prob <- predict(DealLRmodel, DealsOpen[4:16], type = 'response')

DealsOpen$DealProb <- fitOpen.prob
DealsOpen$DealProb <- round(DealsOpen$DealProb,3)
```

```
# Let's export this to excel so we can makes these results more presentable.
write.csv(select(DealsOpen, ID, Name, Owner, DealProb), file = "DealProb.csv")
```

New Data

```
# Cross Validation of model
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)
# Fit the k-folds cross validation model
mod_fit <- train(Win ~ .,  data=Deal.train, method="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)
# Check out the model
print(mod_fit)
```

Validity & Reliability Assessment

```
## Generalized Linear Model
##
## 11079 samples
##    12 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 9971, 9971, 9971, 9971, 9971, 9971, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6775893  0.3538402
```

```
# Use this model to predict on the test set
pred = predict(mod_fit, newdata = Deal.test[-13])
# Look the the confustion matrix of this model
confusionMatrix(data=pred, Deal.test$Win)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 1710  745
##          1  809 1484
##                                           
##                Accuracy : 0.6727          
##                  95% CI : (0.6591, 0.686) 
##     No Information Rate : 0.5305          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.344           
##                                           
##  Mcnemar's Test P-Value : 0.11            
##                                           
##             Sensitivity : 0.6788          
##             Specificity : 0.6658          
##          Pos Pred Value : 0.6965          
##          Neg Pred Value : 0.6472          
##              Prevalence : 0.5305          
##          Detection Rate : 0.3602          
##    Detection Prevalence : 0.5171          
##       Balanced Accuracy : 0.6723          
##                                           
##        'Positive' Class : 0               
## 
```

Random Forest

```r
# The next modeling technique applied is Random Forest.
rfModel <- randomForest(Win ~ ., data=Deal.train, ntree=500, importance=TRUE)
y_pred = predict(rfModel, newdata = Deal.test[-13])
cm = table(Deal.test[,13], y_pred)
cm
```

```
##    y_pred
##        0    1
##   0 2052  467
##   1  612 1617
```

```r
1 - mean(y_pred != Deal.test$Win)
```

```
## [1] 0.7727464
```

```r
confusionMatrix(y_pred, Deal.test$Win)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 2052  612
##          1  467 1617
## 
```

```
##                Accuracy : 0.7727
##                  95% CI : (0.7606, 0.7846)
##     No Information Rate : 0.5305
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5421
##
##  Mcnemar's Test P-Value : 1.166e-05
##
##             Sensitivity : 0.8146
##             Specificity : 0.7254
##          Pos Pred Value : 0.7703
##          Neg Pred Value : 0.7759
##              Prevalence : 0.5305
##          Detection Rate : 0.4322
##    Detection Prevalence : 0.5611
##       Balanced Accuracy : 0.7700
##
##        'Positive' Class : 0
##
```

```
# Predicting the results of the new data using random forest model
y_predOpen = predict(rfModel, newdata = DealsOpen[4:16])
```

New Data

Appendix B

The following section contain the R code used for the data cleaning and visualization used in the project.

Data Cleaning

```
# Count Unique Calls on the Account during the period in which the Opportunity is active
CountCalls <- function(CreDate, CloDate, AccID, OppId){
  numCalls <- subset(dr.task,
                     (dr.task$TaskCreatedDate > CreDate & dr.task$TaskCreatedDate <
                        CloDate)
                     & dr.task$TaskAccountID == AccID)
  return(nrow(numCalls))
}

updDateOppAccCalls <- function()
  for (row in 1:nrow(dr.opportunities)){
    OppID <- dr.opportunities[row, "Opportunity.ID"]
    AccID <- dr.opportunities[row, "Account.ID"]
    CreDate <- dr.opportunities[row, "Created.Date"]
    CloDate <- dr.opportunities[row, "Close.Date"]
    numCalls <- CountCalls(CreDate, CloDate, AccID, OppId)
    dr.opportunities[row, "AccCalls" ] <<- numCalls
  }
# Create an Age in Days field and update it for all opportunities
# Opportunities that are open will use Feb 7th as the day to calculate
# the closed data as this is when the file was received

# Create the new field
dr.opportunities$AgeInDays <- 0
dr.opportunities$AccCalls <- as.integer(dr.opportunities$AccCalls)


# Update the Age of all Closed Opportunities
dr.opportunities$AgeInDays <-
  ifelse((dr.opportunities$Stage == 'Closed Lost') | (dr.opportunities$Stage ==
                                                        'Closed Won'),
         difftime(dr.opportunities$Close.Date, dr.opportunities$Created.Date,
                  units="days"),
         dr.opportunities$AgeInDays)

#Update the Age of all Open Opportunities with Feb 7th date
dr.opportunities$AgeInDays <-
  ifelse((dr.opportunities$Stage != 'Closed Lost') & (dr.opportunities$Stage !=
                                                        'Closed Won'),
         round(difftime("2018-02-07", dr.opportunities$Created.Date,
                        units="days"),0),
         dr.opportunities$AgeInDays)

# Import User Data
dr.user<- read.csv('User.csv', header = FALSE)


colnames(dr.user) <- c('UserID','Username', 'LastName','FirstName','ProfileName',
                       'RoleName','Department', 'ManagerFullName',
                       'Active','CreatedDate', 'UserTenure','UserTotalCalls',
                       'DailyCallAvg')

dr.user$UserID <- as.character(dr.user$UserID)

# Convert Date to Date format
dr.user$CreatedDate <- as.Date(dr.user$CreatedDate)

#Create and Update User Tenure Column
dr.user$UserTenure2 <- 0
dr.user$UserTenure2 <- round(difftime("2018-02-07", dr.user$CreatedDate,
                                      units="days"),0)

#Create and update a todal calls column
dr.user$UserTotalCalls2 <- 0

# Function to count all calls in Tasks user by user
UserCalls <- function(UserID){
  numCalls <- subset(dr.task, dr.task$TaskCreatedByID == UserID)
  return(nrow(numCalls))
}

updUserCalls <- function()
  for (row in 1:nrow(dr.user)){
    UserID <- dr.user[row, "UserID"]
    dr.user[row, "UserTotalCalls2" ] <<- UserCalls(UserID)
}

# Call the function to update the User's total calls
updUserCalls()

# All the opportunity calls - calls where the opportunity was indicated
# The WhatID of the task = opportunityID

# Create the new column

dr.opportunities$OppCalls <- 0

CountCalls <- function(OppID){
  numCalls <- subset(dr.task, dr.task$TaskWhatID == OppID)
  return(nrow(numCalls))
}

updOppCalls <- function()
  for (row in 1:nrow(dr.opportunities)){
    OppID <- dr.opportunities[row, "Opportunity.ID"]
    dr.opportunities[row, "OppCalls" ] <<- CountCalls(OppID)
}

updOppCalls()

# Opportunity calls
# Group all the tasks by TaskWhatID (this related to the opportunityID in
# Opportunity)
call_count <- dr.task %>% group_by(TaskWhatID) %>% summarise( count_calls = n())

#merge the dataframes

colnames(call_count) <- c('Opportunity.ID', 'TotOpCalls')

dr.opportunities <- merge(x = dr.opportunities, y = call_count, by =
                            "Opportunity.ID", all.x = TRUE)

# User
dr.user.sub <- dr.user[,c("UserID","ProfileName","Active","UserTenure",
                          "UserTotalCalls2","UserTenure2")]
colnames(dr.user.sub) <- c("User.ID","ProfileName","UserActive","UserTenure",
                           "UserTotalCalls","UserTenure")
dr.user.sub$UserTotalCalls <- as.integer(dr.user.sub$UserTotalCalls)
dr.user.sub$UserTenure <- ROUND(as.integer(dr.user.sub$UserTenure),0)

#merge dataframe opportunities with user

dr.opportunities <- merge(x = dr.opportunities, y = dr.user.sub, by = "User.ID",
                          all.x = TRUE)

dr.opportunities$OppCalls <- NULL
dr.opportunities$UserTenure <- NULL

dr.opportunities.back <- dr.opportunities

colnames(dr.opportunities)[25] <- c("UserTenure")
ncol(dr.opportunities)

# Load Accounts

dr.account <- read.csv('Accounts.csv')
dr.account$Created.Date <- as.Date(dr.account$Created.Date,'%d/%m/%Y')

dr.account$AccountAge <- as.integer(round(difftime("2018-02-07",
                                                   dr.account$Created.Date, units =
                                                     "days"),0))

dr.opportunities$UserTenure <- as.integer(dr.opportunities$UserTenure)

dr.account.sub <- dr.account[,c("Account.ID","Billing.State.Province","AccountAge")]
#Merge into opps
dr.opportunities <- merge(x = dr.opportunities, y = dr.account.sub, by =
                            "Account.ID", all.x = TRUE)

#Contacts
dr.contact <- read.csv('Contacts.csv')
#Count contacts at Account
contactCount <- dr.contact %>% group_by(Account.ID) %>% summarise( NumContacts = n())
#Merge into opportunities
```

```r
dr.opportunities <- merge(x = dr.opportunities, y = contactCount, by = "Account.ID",
                          all.x = TRUE)

unique(dr.opportunities$Billing.State.Province)
str(dr.opportunities)

#Change na to 0 for the column
dr.opportunities$TotOpCalls <- ifelse(is.na(dr.opportunities$TotOpCalls),0,
                                      dr.opportunities$TotOpCalls)
dr.opportunities$NumContacts <- ifelse(is.na(dr.opportunities$NumContacts),0,
                                       dr.opportunities$NumContacts)

#Add column for call average per day for user
dr.opportunities$UserDayCallAvg <- round(dr.opportunities$UserTotalCalls/
                                           dr.opportunities$UserTenure,2)

#Count Emails
# Add a column for all email Tasks
dr.task$EmailCount <- ifelse(grepl("mail",dr.task$TaskType),1,0)

# Count all 'Email Calls' that are Opportunity Calls & Count all 'Email Calls'
# among account calls
# Update EmailCalls - Opportunity calls that were emails

EmailCount <- dr.task %>% group_by(TaskWhatID) %>% summarise( NumOppEmails =
                                                                sum(EmailCount))

colnames(EmailCount) <- c("Opportunity.ID","NumOppEmails")


dr.opportunities <- merge(x = dr.opportunities, y = EmailCount, by =
                            "Opportunity.ID",
                          all.x = TRUE)

dr.opportunities$NumOppEmails <- ifelse(is.na(dr.opportunities$NumOppEmails),0,
                                        dr.opportunities$NumOppEmails)
dr.opportunities$UserDayCallAvg <- ifelse(is.na(dr.opportunities$UserDayCallAvg),0,
                                          dr.opportunities$UserDayCallAvg)

# Now for all associated accounts how many calls were emails
EmailAccountCall <- dr.task %>% group_by(TaskAccountID) %>%
  summarise( NumAccountEmails = sum(EmailCount))

colnames(EmailAccountCall) <- c("Account.ID","NumAccountEmails")
dr.opportunities <- merge(x = dr.opportunities, y = EmailAccountCall, by =
                            "Account.ID",
                          all.x = TRUE)
dr.opportunities$NumAccountEmails <- ifelse(is.na(dr.opportunities$NumAccountEmails),0,
                                            dr.opportunities$NumAccountEmails)

#Create function to update Email Call count
#Load Countries
dr.CountryAcc <- read.csv('AccountCountry.csv')
```

```
#merge
dr.opportunities <- merge(x = dr.opportunities, y = dr.CountryAcc, by =
                            "Account.ID", all.x = TRUE)

dr.opportunities$Billing.Country <-  ifelse(grepl(c(
  'On|qc|CANADA|n.s.|ON|BC|QC|PQ|SK|MB|PE|NB|NS|AB|NL|Ontario|Alberta|Quebec|
  Manitoba|Nova Scotia|Saskatoon|alberta'),dr.opportunities$Billing.State.Province),1,0)

dr.opportunities$Win <- ifelse(dr.opportunities$Stage == 'Closed Won',1,0)
dr.opportunities$Billing.Country <- as.factor(dr.opportunities$Billing.Country)

# Correlation of all factors
corrgram(boxData, order=TRUE, lower.panel = panel.shade,
         upper.panel = panel.cor, text.panel = panel.txt)


vars2 <- c('Revenue.Potential', 'AccCalls', 'TotOpCalls',
           'UserTotalCalls', 'UserTenure', 'NumContacts', 'NumOppEmails',
           'NumAccountEmails','Win')
boxData.cor <- cor(boxData[,vars2], use='pair')
boxData.eig <- eigen(boxData.cor)$vectors[,1:2]
e1 <- boxData.eig[,1]
e2 <- boxData.eig[,2]


plot(e1,e2,col='white', xlim=range(e1,e2), ylim=range(e1,e2))
text(e1,e2, rownames(boxData.cor), cex=.9)
title("Eigenvector plot of D&R data")
arrows(0, 0, e1, e2, cex=0.5, col="red", length=0.1)
```

Visualization

```
filter(dr.opportunities, Stage == 'Closed Lost' | Stage == 'Closed Won' ) %>%
  group_by(Stage) %>%
  summarise(avg=mean(AgeInDays)) %>% ggplot(aes(x=Stage, y=avg)) +
  geom_bar(stat = 'identity', aes(fill=Stage), alpha=.5) +
  theme_wsj(base_family = "serif") + theme(axis.title=element_text(size=12)) +
  scale_colour_wsj() +
  scale_fill_wsj() +
  theme(legend.title=element_blank()) +
  ylab('Average Age in Days of Opportunity') + ggtitle('Age of Won vs. Lost Deals') +
  xlab("") + scale_x_discrete(labels=c("Lost","Won")) +
  annotate(geom="text", x = 2, y = 100, label =
               'The age of winning deals are almost \n  twice that of losing deals',
             color='black', fontface=2, size = 4,   family="Comic Sans MS")   +
  theme(legend.position="none")

OppAgeChart <- filter(dr.opportunities, (Stage == 'Closed Lost' | Stage ==
                                           'Closed Won') &
                        AgeInDays > 0 ) %>%
  select(AgeInDays, Stage)
```

```
# Order by Days
OppAgeChart <- arrange(OppAgeChart, AgeInDays )

# Add Period column
OppAgeChart$period <- ceiling(OppAgeChart$AgeInDays/30)
OppAgeChart$Count <- 1

OppAgeChart$period <- ifelse(OppAgeChart$period == 0,1,OppAgeChart$period)
OppAgeChart2 <- OppAgeChart %>% group_by(period, Stage) %>% summarise(Count =
                                                              sum(Count)) %>%
  spread(Stage, Count)
View(OppAgeChart2)
colnames(OppAgeChart2) <- c('Period','Lost','Won')


OppAgeChart2$PecentWon <-  OppAgeChart2$Won / (OppAgeChart2$Won +
                                                 OppAgeChart2$Lost)
OppAgeChart2$PecentWon <- round(OppAgeChart2$PecentWon,2) *100
OppAgeChart2$PecentLost <-  OppAgeChart2$Lost / (OppAgeChart2$Won +
                                                   OppAgeChart2$Lost)
OppAgeChart2$PecentLost <- round(OppAgeChart2$PecentLost,2) *100

OppAgeChart3 <- subset(OppAgeChart2, OppAgeChart2$Period<10)

pl <- ggplot(data=OppAgeChart3, aes(x=Period)) + geom_line(color='yellow',
                                                           size = 3, aes(y=PecentWon))
#geom_line(color='red', size = 3, aes(y=PecentLost)) +

pl <- pl +  theme_wsj(base_family = "serif") + theme(axis.title=element_text(size=12))
pl <- pl +    theme(legend.title=element_blank()) + xlab('Age of Deal in Days')
pl <- pl +      ylab('Percent Won') + ggtitle('% of Deals Won by Time Period')
pl <- pl +   geom_hline(yintercept=45, color = "light blue", size=3)
#pl <- pl +    geom_text(aes(0,43,label = "AVERAGE"), color="light blue", size=5)
pl <- pl +  scale_x_continuous(breaks = 1:10,labels =
                       c("0-30", "30-60", "60-90", "90-120", "120-150",
                         "150-180", "180-210","210-240","240-270","270-300"))
pl <- pl + annotate("text", x = 1.3, y = 48, label = 'Average', color='light blue',
                    fontface=2, size = 6, family="Comic Sans MS")
pl <- pl + annotate(geom="text", x = 5.5, y = 55, label =
          'In the first 70 days deals are more likely to be won \n than lost,
          the opposite is true thereafer',
          color='black', fontface=2, size = 4,   family="Comic Sans MS")


# In the first 70 days deals are more likely to be won than lost, the opposite is
# true thereafer.

# Graph 3
OppAgeChart2$Lost <- ifelse(is.na(OppAgeChart2$Lost),0,OppAgeChart2$Lost)
OppAgeChart2$Won <- ifelse(is.na(OppAgeChart2$Won),0,OppAgeChart2$Won)
windowsFonts()
TotalWonDeals <- sum(OppAgeChart2$Won)
TotalLostDeals <- sum(OppAgeChart2$Lost)
```

```
OppAgeChart2$ShareWon <- (OppAgeChart2$Won / TotalWonDeals) * 100
OppAgeChart2$ShareLost <- (OppAgeChart2$Lost / TotalLostDeals) * 100

pl <- ggplot(data=OppAgeChart2, aes(x=Period)) + geom_line(color='green', size = 2,
                                                          aes(y=ShareWon), alpha=0.5)
pl <- pl +  theme_wsj(base_family = "serif") + theme(axis.title=element_text(size=12))
pl <- pl + geom_line(color='red',  alpha=0.5, size = 3, aes(y=ShareLost))
pl <- pl +    theme(legend.title=element_blank()) + xlab('Age in Months')
pl <- pl +      ylab('Share of Deals') + ggtitle('Share of Won and Lost \n
                                                 Deals by Month')
pl <- pl + annotate(geom="text", x = 20, y = 15, label =
           'Over 50% of won deals are closed within 60 days, \n 27% of lost deals are
           still open after 5 months ',
         color='black', fontface=2, size = 4,   family="Comic Sans MS")
pl <- pl + annotate(geom="text", x = 13, y = 26, label =
                      '27 % of winning deals are won \n in the first month ',
                    color='black', fontface=2, size = 4,   family="Comic Sans MS")

print(pl)

# Boxplot of Department
dr.opportunities %>% select(ProfileName, Win) %>%

## GRAPH NEW
select(dr.opportunities,Opportunity.Owner..Department,Win) %>%
  group_by(Opportunity.Owner..Department) %>%
  summarise(mean=mean(Win)*100) %>% filter(mean!=0) %>% arrange(desc(mean)) %>%
  ggplot(aes(x=reorder(Opportunity.Owner..Department,-mean),y=mean)) +
  geom_bar(stat = "identity", aes(fill=Opportunity.Owner..Department),alpha=0.5) +
  theme_wsj(base_family = "serif") + theme(axis.title=element_text(size=12)) +
  theme(axis.text.x= element_text(size=9, face=1)) +
  ggtitle("% Deals won by Department") +
  theme(legend.title=element_blank()) + xlab('Department') +
  ylab('Percent Deals Won') +
  theme(axis.text.x=element_text(angle = 90, vjust=0.5)) +
  theme(legend.position="none") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 12)) +
  annotate(geom="text", x = 8.5, y = 53, label =
             'Win Rates vary greatly among departments,  \n from over 60% for
           Quebec Freight to less than \n half that for USA Freight ',
           color='black', fontface=2, size = 4,   family="Comic Sans MS")

# Boxplots for comparison of Won opportunities to Lost ones - looking at factors
boxData <-
select(dr.opportunities, Stage, Revenue.Potential, AccCalls, TotOpCalls, Win,
       UserTotalCalls, UserTenure, NumContacts, NumOppEmails, NumAccountEmails,
       UserDayCallAvg, Billing.Country ) %>%
  filter(Stage == 'Closed Lost' | Stage == 'Closed Won' )

boxData$Billing.Country <- as.numeric(boxData$Billing.Country)

Call.Avg.plot <- ggplot(data=boxData, aes(x=Stage, fill=Stage, alpha=0.4)) +
  geom_boxplot(aes(y=UserDayCallAvg)) +
  theme_wsj() +
  theme(legend.position="none") + scale_x_discrete(labels=c("Lost","Won")) +
  scale_colour_economist() + scale_fill_tableau() +
  ggtitle("Won vs. Lost Call Averages") +
  theme(plot.title = element_text(size=12))

Call.Tenure.plot <- ggplot(data=boxData, aes(x=Stage, fill=Stage, alpha=0.4)) +
  geom_boxplot(aes(y=UserTenure)) +
  theme_wsj() +
  theme(legend.position="none") + scale_x_discrete(labels=c("Lost","Won")) +
  scale_colour_economist() + scale_fill_tableau() +
  ggtitle("Won vs. Lost User Tenure") +
  theme(plot.title = element_text(size=12))


Call.OppCalls.plot <- ggplot(data=boxData, aes(x=Stage, fill=Stage, alpha=0.4)) +
  geom_boxplot(aes(y=TotOpCalls)) +
  theme_wsj() +
  theme(legend.position="none") + scale_x_discrete(labels=c("Lost","Won")) +
  scale_colour_economist() + scale_fill_tableau() +
  ggtitle("Won vs. Lost Deal Deal Calls") +
  theme(plot.title = element_text(size=12))

Call.NumCon.plot <- ggplot(data=boxData, aes(x=Stage, fill=Stage, alpha=0.4)) +
  geom_boxplot(aes(y=NumContacts)) +
  theme_wsj() +
  theme(legend.position="none") + scale_x_discrete(labels=c("Lost","Won")) +
  scale_colour_economist() + scale_fill_tableau() +
  ggtitle("Won vs. Lost # Contacts") +
  theme(plot.title = element_text(size=12))

grid.arrange(Call.Avg.plot, Call.Tenure.plot,
             Call.OppCalls.plot, Call.NumCon.plot, ncol=2)

#oppUsers$CloseDate <- as.yearqtr(oppUsers$Close.Date)
oppUsers$CloseDate <- format(oppUsers$Close.Date,"%Y")

oppUsers <- oppUsers %>% group_by(format(oppUsers$Close.Date,"%Y"),
                                  Opportunity.Owner) %>%
  summarise(UserTenure=max(UserTenure), Revenue=sum(Revenue.Potential),
            Num=n(), Wins=sum(Win))

colnames(oppUsers)[1] <- c("CloseDate")
str(oppUsers)
oppUsers$WinRate <- round(oppUsers$Wins/oppUsers$Num,2)
View(oppUsers)

ggplot(data=oppUsers, aes(x=CloseDate,y=WinRate)) +
  geom_line(aes(color=Opportunity.Owner,size=Revenue))


top10Owners <- oppUsers %>% group_by(Opportunity.Owner) %>%
  summarise(TotRev=sum(Revenue)) %>% arrange(desc(TotRev)) %>% top_n(n=6, wt=TotRev)
```

```r
#Merge with original

top10Owners$TotRev <- NULL

oppUsersNew <- merge(x =oppUsers, y = top10Owners, by = "Opportunity.Owner",
                     all.x = FALSE)

ggplot(data=oppUsersNew, aes(x=as.numeric(CloseDate),y=Revenue)) +
  geom_point(aes(color=Opportunity.Owner,size=WinRate), alpha=0.6) +
  theme_wsj(base_family = "serif") +  theme(axis.title=element_text(size=12)) +
  theme(axis.text.x= element_text(size=9, face=1)) + scale_color_wsj() +
  ggtitle("Top 6 Reps Win Rate and Revenue") +
  theme(legend.title=element_blank()) + xlab('Year') + ylab('Potential Revenue') +
  theme(plot.title = element_text(size=18)) +
  scale_y_continuous(breaks = c(0,2500000,5000000,7500000),labels =
                       c("0", "2.5 M", "5 M", "7.5 M")) +
  annotate(geom="text", x = 2014, y = 6600000, label =
             'Winning Reps usually stay winning,\n the color of the dot is the Rep,
           \nits size is the win rate ',
           color='black', fontface=2, size = 3.5,   family="Comic Sans MS") +
  guides(colour = guide_legend(override.aes = list(size=7))) +
  geom_line(aes(color=Opportunity.Owner),size=2,alpha=.2)

# Deparment
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~

oppDept <- select(dr.opportunities, Close.Date, Opportunity.Owner..Department,
                  Revenue.Potential, Win )

# Revenue to 0 if did not win
oppDept$Revenue.Potential <- ifelse(oppDept$Win==0,0,oppDept$Revenue.Potential)
View(oppDept)




oppDept <- oppDept %>% group_by(format(oppDept$Close.Date,"%Y"),
                                Opportunity.Owner..Department) %>%
  summarise(Revenue=sum(Revenue.Potential), Num=n(), Wins=sum(Win))

colnames(oppDept)[1] <- c("CloseDate")
colnames(oppDept)[2] <- c("Dept")

oppDept$WinRate <- round(oppDept$Wins/oppDept$Num,2)
View(oppDept)


ggplot(data=subset(oppDept,oppDept$Dept!=""), aes(x=as.numeric(CloseDate),y=Revenue)) +
  geom_point(aes(color=Dept,size=WinRate), alpha=0.6) +
  theme_wsj(base_family = "serif") +  theme(axis.title=element_text(size=12)) +
  theme(axis.text.x= element_text(size=9, face=1)) +
  ggtitle("Department Win Rate and Revenue") +
  theme(legend.title=element_blank()) + xlab('Year') + ylab('Potential Revenue') +
```

```
  theme(plot.title = element_text(size=18)) +
  scale_y_continuous(breaks = c(0, 1e+07, 2e+07, 3e+07),
                     labels = c("0", "10 M", "20 M", "30 M")) +
  annotate(geom="text", x = 2013.5, y = 26000000, label =
             'Winning Departments usually stay winning,\n
the color of the dot is the Dept.,
           \n its size is the win rate ',
           color='black', fontface=2, size = 3.5,   family="Comic Sans MS") +
  guides(colour = guide_legend(override.aes = list(size=7))) +
  geom_line(aes(color=Dept),size=2,alpha=.2)
```