Bazeley, Mikiko
Springboard 2018

Report:
Predicting Outcomes of Demo Calls for a SaaS Sales Oriented Company

Introduction:

For my first capstone, my goal was to predict the outcome of my company's sales demo calls. Predicting the outcome of the first customer engagement is a task that can help a company forecast sales and understand the drivers of sales qualification.

Data was collected from my company's internal data warehouse by querying our Salesforce tables (specifically Leads and Intro Calls) and was then cleaned and processed. Exploratory analysis consisted of identifying potential features and applying various statistical tests to determine strength & direction of relationship with the outcome variable.

After exploratory analysis a master dataset was created from merging the two starting data sets and I used a variety of classic machine learning classifiers to predict intro call outcomes.

Through a combination of model selection and hyperparameter tuning an accuracy of 79.3% was achieved, showing that machine learning can provide value in the business setting with assisting forecasting and reporting tasks.

Key Stakeholders:

Potential parties that could be interested in this project include:

1. Current company sales leadership
2. Current company marketing leadership
3. Other revenue operations professionals wanting to understand how to conduct similar analysis & projects

Data Overview:

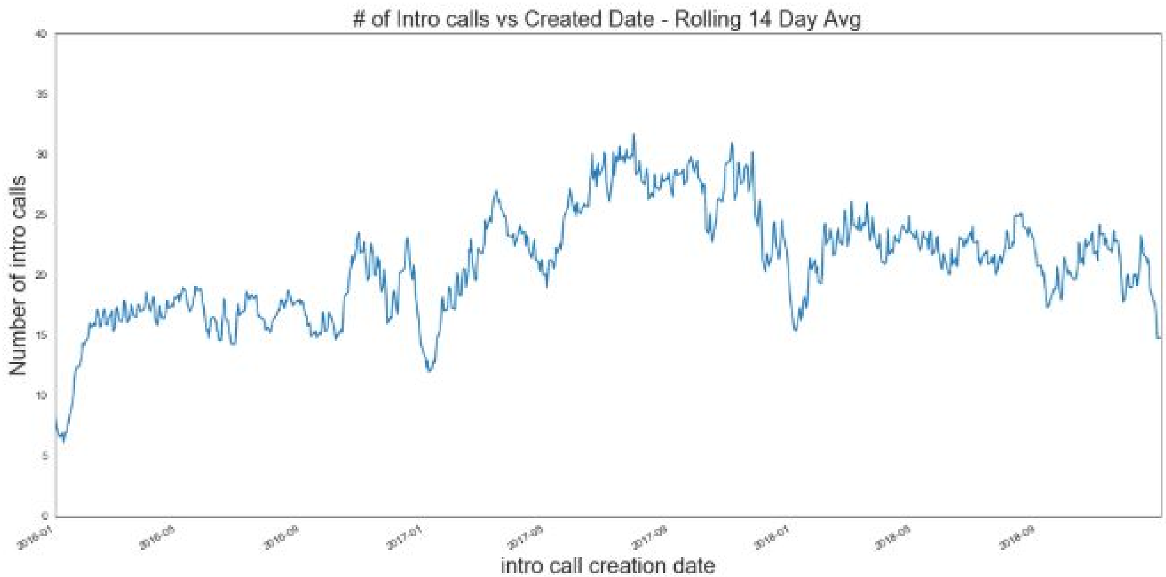
Data for the project came from our AWS Redshift data warehouse, which includes tables that perfectly mirror the Salesforce objects and fields (and include columns documenting these relationships). Data was included from Jan 1, 2016 to Jan 1, 2019.

Examples of the objects and fields include:

Leads Table/Object	Intro Call Table/Custom Object
● Id ● First Name ● Last Name ● Company ● Job Title ● Email ● Landing Page ● etc	● Id ● Lead Id ● Lead Name ● Company Name ● Assigned to ● Created Date ● Scheduled Call Date ● etc

The Lead table had 493K entries with 108 columns while the Intro Calls table had 23K entries with 115 columns. Data types included dates, numeric, currency based, text, picklist/categorical, etc. Columns could also be duplicated across entities like the Intro Call and Leads tables. Given the Leads and Intro Call data sets also contained in progress prospects, I needed to focus on Intro Calls that had been closed out.

Of 22.9K Intro Call records, 12.8K (56%) were qualified vs 10K (44%) disqualified indicating fairly well balanced classes. Each row in the Intro Call table represents a single instance of a call but multiple rows in the Lead table could represent a single person, however, only a single Lead record will be attached to an Intro Call record.



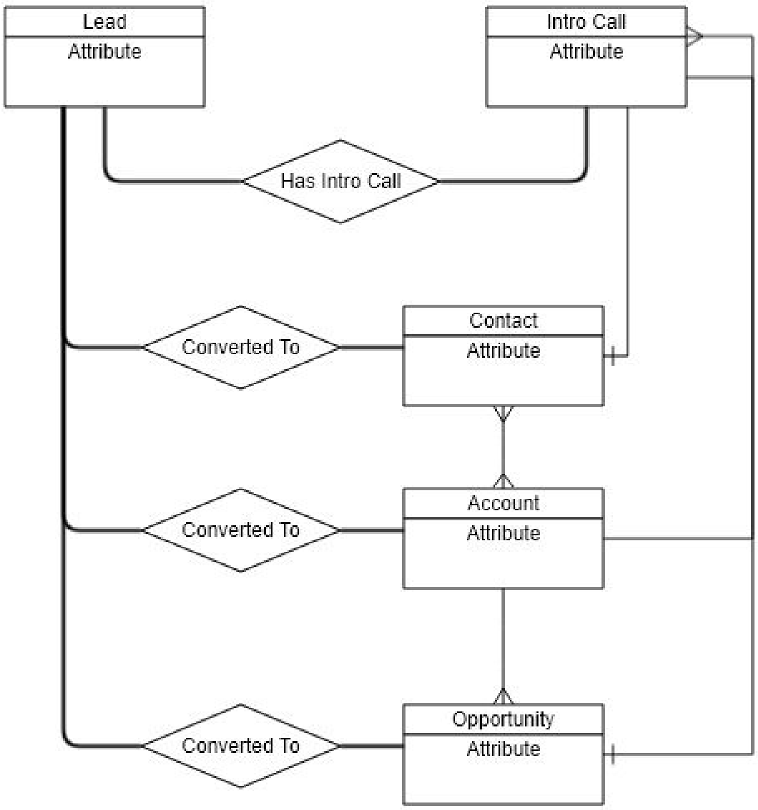
Some important considerations needed to be handled given the data quality & source.

1) Multiple intro calls for the same individual could have occurred throughout the time where the first call was scored as disqualified but the second call was then qualified.
2) A significant number of fields were duplicated across tables - due to the quasi relational structure of Salesforce, objects could be mimicked as tables but fields needed to be duplicated for the Salesforce user
3) Not all fields were used throughout time & the values didn't always reflect the field's stated purpose
4) Because of the tribal knowledge nature of the company, some fields could reflect data leakage without the business always knowing.

As a result of the prior considerations, careful mapping and many iterations of consultation with the relevant business owners needed to occur.

Data Acquisition & Processing:

Using sqlalchemy, psycopg2 (a postgreSQL driver) I queried our data warehouse for the two tables. Once the tables were queried into pandas data frames I then merged them via a common identifier and proceeded with intensive data scrubbing.



One step of cleaning involved converting all the datetime fields to a common type. For example: ‘2018-11-08T20:12:05.000Z’, ‘Q1-2015’, ‘10/17/2014 17:09’, ‘10/28/2014’ are just four examples of the 10+ data columns that needed to be parsed and converted to a datetime object. I wrote a function that would take a dataframe, the target column to be parsed, name of a new column, and the date time pattern to pass in. The function clean_dates takes the specified columns, parses the timedate string, returns a datetime object as the new column and deletes the old column.

The next import step was re-grouping the categorical features. After inspecting the different grouped columns and values, I remapped the values using dictionaries containing the new group values and deleted the old columns.

Fields with long text data were also dropped due to inexperience with NLP techniques. Additional columns were created in order to measure the duration between various stages of leads, demos, and opportunities.
After creating all the necessary columns for exploratory analysis, additional id and demographic columns were deleted.

Exploratory Analysis:

After data cleaning and processing I then analyzed a sample of potential key drivers of call qualifications. They included:

- inferScore___Lead_AddedInfo
- totalEMails___Lead_AddedInfo
- totalCalls___Lead_AddedInfo
- introCallCreated_leadCreated_delta
- assignedToRole___IntroCall_OtherInfo_map
- country___Lead_LeadCompanyInformation_map
- trafficChannel___Lead_MarketingInformation_map_map

- product2___IntroCall_MeetingDetails_WalkMe

After visual inspection and statistical testing, three feature seemed promising in predicting qualification, specifically ‘Total Calls & Emails’, ‘Lead Score’, and ‘Intro Call Creation Delta’ (or freshness of the lead).

Table Summary:

Variable	Summary of Visual Inspection	Summary of Statistical Inspection
• Intro Call Creation Date (Month, Year)	No clear pattern	
• Marketing Channel	Difficult to discern but could be a driver of qualification	
• Customer Type	Could be a driver of qualification	
• Country	Difficult to discern, would need to perform goodness-of-fit test potentially	
• Landing Page	Difficult to discern, would need to perform goodness-of-fit test potentially	
• Total Calls & Emails		• Permutation Test: ○ P-val: 0.0000 • Bootstrap Test: ○ P-val: 0.0000 • Mann-Whitney: ○ P-val: 1.3737e-57 • Welch’s T-Test: ○ P-val: 3.9676e-90
• Lead Score		• Permutation Test: ○ P-val: 0.0000 • Bootstrap Test: ○ P-val: 0.0000 • Mann-Whitney: ○ P-val: 1.31465e-107 • Welch’s T-Test: ○ P-val: 6.33020e-99
• Intro Call Creation Delta		• Permutation Test: ○ P-val: 0.1572 • Bootstrap Test: ○ P-val: 0.0788 • Mann-Whitney: ○ P-val: 1.5869e-05 • Welch’s T-Test: ○ P-val: 0.16128627

Feature Engineering:

Features that were created include:

- 'introCallCreated_leadCreated_delta'

Features that were cleaned/recategorized include:

- 'trafficChannel___Lead_MarketingInformation'
- 'Target___IntroCall_Outcome'
- 'customerOrEmployee___IntroCall_MeetingDetails'
- 'rescheduled___IntroCall_Outcome_map'
- 'createdDayOFWeek___IntroCall_AddedInfo_map'
- 'pitch___Lead_AddedInfo_map'
- 'decisionMaker___IntroCall_MeetingDetails_map'
- 'employeeCategory___Lead_AddedInfo_map'
- 'rejectedReason___IntroCall_Outcome_map'
- 'status___Lead_ConversionInformation_map'
- 'trafficChannel___Lead_MarketingInformation_map'
- 'statusReason___Lead_ConversionInformation_map'

Feature Selection:

Given the poor quality of the data set and complexity of the challenge, I decided to use the following 37 features (out of the available 108 (Leads) & 115 (Intro Calls):

- 'rescheduledFromIntroCall___IntroCall_Outcome'
- 'doubleDipper___IntroCall_MeetingDetails'
- 'projectDueQ___IntroCall_MeetingDetails'
- 'produce1___IntroCall_MeetingDetails'
- 'product2___IntroCall_MeetingDetails'
- 'appType___IntroCall_MeetingDetails'
- 'newOrExistingCustomer___IntroCall_LeadInformation'
- 'introCallMktSource___IntroCall_LeadInformation'
- 'title___Lead_PersonalInformation'
- 'customerType___Lead_LeadCompanyInformation'
- 'company___Lead_LeadCompanyInformation'
- 'city___Lead_LeadCompanyInformation'
- 'state___Lead_LeadCompanyInformation'
- 'country___Lead_LeadCompanyInformation'
- 'linkedinPage___Lead_MarketingInformation'
- 'mktChannelcampaign___Lead_MarketingInformation'
- 'landingPage___Lead_MarketingInformation'
- 'landingPageUrl___Lead_MarketingInformation'
- 'duplicateLead___Lead_ImportantSystemInfo'

- 'isconverted___Lead_ImportantSystemInfo'
- 'isdeleted___Lead_ImportantSystemInfo' 'inferScore___Lead_AddedInfo'
- 'totalCalls___Lead_AddedInfo' 'totalEMails___Lead_AddedInfo'
- 'usersAmount___Lead_AddedInfo' 'marketingCampaignID___Lead_AddedInfo'
- 'marketingChannelAdID___Lead_AddedInfo'
- 'converteddate___Lead_ConversionInformation_clean'
- 'createddate___Lead_ImportantSystemInfo_clean'
- 'createddate___IntroCall_ImportantSystemInfo_clean'
- 'introCallCreated_leadCreated_delta'
- 'rescheduled___IntroCall_Outcome_map'
- 'customerOrEmployee___IntroCall_MeetingDetails_map'
- 'createdDayOFWeek___IntroCall_AddedInfo_map' 'pitch___Lead_AddedInfo_map'
- 'decisionMaker___IntroCall_MeetingDetails_map'
- 'employeeCategory___Lead_AddedInfo_map' 'Target___IntroCall_Outcome_map'
- 'rejectedReason___IntroCall_Outcome_map'
- 'status___Lead_ConversionInformation_map'
- 'trafficChannel___Lead_MarketingInformation_map'
- 'statusReason___Lead_ConversionInformation_map'

Model Selection & Performance:

Given this was my first project I decided to use some of the classic machine learning classifiers (Logistic Regression, Random Forest classifier, and a Gradient Boosted Trees).

Ultimately the best performing model turned out to be the Random Forest classifier and Gradient Boosted tree model, resulting in an accuracy of 80% with additional hyperparameter tuning.

The top 5 features across all three models in determining Intro Call Qualification Status included:

- inferScore___Lead_AddedInfo
- totalEMails___Lead_AddedInfo
- totalCalls___Lead_AddedInfo
- introCallCreated_leadCreated_delta
- assignedToRole___IntroCall_OtherInfo_map

The features I assumed would be highly ranked but weren't included:

- country___Lead_LeadCompanyInformation_map
- trafficChannel___Lead_MarketingInformation_map_map
- product2___IntroCall_MeetingDetails_WalkMe

Interestingly the top performing model also departed from the other models in terms of features ranked #5-#10 (see screenshot below).

Summary of approaches used and performance:

Model	**Performance**	**Performance with Param. Tuning**	**Optimal Params**	**Top Features**
Logistic Regression	73%	73%	{'C': 1, 'max_iter': 100}	
Random Forest	74%	80.0%	{'bootstrap': False, 'max_depth': 60, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 400}	1. inferScore___Lead_AddedInfo 2. totalCalls___Lead_AddedInfo 3. totalEMails___Lead_AddedInfo 4. introCallCreated_leadCreated_delta 5. assignedToRole___IntroCall_OtherInfo_map 6. product2___IntroCall_MeetingDetails_WalkMe for Salesforce - Web 7. mnth_createddate___IntroCall_ImportantSystemInfo_clean 8. mnth_createddate___Lead_ImportantSystemInfo_clean 9. trafficChannel___Lead_MarketingInformation_map_map 10. customerOrEmployee___IntroCall_MeetingDetails_map
Gradient Boosted	78%	80%	{'colsample_bytree': 1.0, 'gamma': 1, 'max_depth': 6, 'min_child_weight': 9, 'subsample': 1.0}	1. totalEMails___Lead_AddedInfo 2. introCallCreated_leadCreated_delta 3. customerType___Lead_LeadCompanyInformation_map 4. inferScore___Lead_AddedInfo 5. year_createddate___IntroCall_ImportantSystemInfo_clean 6. totalCalls___Lead_AddedInfo 7. product2___IntroCall_MeetingDetails_WalkMe for Salesforce - Web 8. customerOrEmployee___IntroCall_MeetingDetails_map 9. year_createddate___Lead_ImportantSystemInfo_clean 10. product2___IntroCall_MeetingDetails_WalkMe for SharePoint

	Random Forest - 1 Hot	Random Forest - Not Hot	Gradient Boosted - 1 Hot	Gradient Boosted - Not Hot
inferScore___Lead_AddedInfo	1	1	3	5
totalEMails___Lead_AddedInfo	2	2	2	1
totalCalls___Lead_AddedInfo	3	3	6	3
introCallCreated_leadCreated_delta	4	4	4	2
assignedToRole___IntroCall_OtherInfo_map	5	5		7
mnth_createddate___IntroCall_ImportantSystemInfo_clean	6			
mnth_createddate___Lead_ImportantSystemInfo_clean	7	7		10
country___Lead_LeadCompanyInformation_map	8	10		
trafficChannel___Lead_MarketingInformation_map_map	9	9		
product2___IntroCall_MeetingDetails_WalkMe	10		5	
mnth_createddate___IntroCall_ImportantSystemInf		6	8	
year_createddate___IntroCall_ImportantSystemInf		8	1	
customerType___Lead_LeadCompanyInformation_map			7	6
year_createddate___Lead_ImportantSystemInfo_clean			9	
customerOrEmployee___IntroCall_MeetingDetails_map			10	8
year_createddate___IntroCall_ImportantSystemInf...				4
decisionMaker___IntroCall_MeetingDetails_map				9

Take-Aways:

My biggest takeaways from the project given the feature ranking & exploratory analysis:

- Quality of lead can continue to impact downstream qualification & sales process beyond the Marketing to Sales Handoff, with lead scores differing between Qualified & Not Qualified Intro Calls (Lead Score).
- Level of engagement can be an important indicator of a qualified prospect & can show inefficiencies in engagements with not qualified prospects (Total Calls + Emails).
- Freshness of lead doesn't seem to impact outcome (Lead to Intro Call Creation Delta).

Given the results of the models and tests, my project based recommendations are:

- Given that disqualified intro calls were correlated with higher calls & emails, one possible suggestion could be to train the sales teams to front load discovery questions for earlier disqualification.
- Given also the difference in distributions of lead scores by qualified vs. disqualified, there could be downstream impact from marketing letting in poorer quality leads. Some of their assertions should also be evaluated as it seems their leads aren't as high quality as expected.

For future work, using the model building process developed in this project, we can work with the sales org to ask similar questions:

- Can we predict what deals will close?
- Can we predict which existing customers are candidates for upsell/cross sell?
- How can we utilize the insights from both the intro call qualification process to improve internal processes around data collection?
- Once these labels are predicted, how can we then improve our forecasts?

My recommendations for the organization:

- Create tighter communication with upstream application teams so the data warehouse matches the state of the data providers.
- Constant re-evaluation and assessment of data points being collected should be done in order to triage data quality issues before analytical projects need to be started.
- Significant data cleaning and extraction time could have been saved by the presence of relevant documentation, especially around data owners.