Analysis Of Sales Opportunities

Anna Mellema

A report submitted for the course
CMSE 202 - Computational Modeling and Data Analysis
Michigan State University

April 2021

Chapter 1

Background and Motivation

This part of the paper is meant to give the reader the necessary background for the project and describe why I chose to do this project. Section 1.1 gives the background material necessary in order to read this report, Section 1.2 gives my motivation for pursuing this question, and related work that I have done is given in Section 1.3.

1.1 Background

Currently, I intern at a company called Service Express. We are a Grand Rapids based company that does Data Center maintenance. This means we provide maintenance for companies on their Server, Storage, and Network equipment. At the time of this project, we have been working on creating changes in our Sales department, with the goal of increasing our profit. One of these changes is shifting to a focus on growing our pipeline. Pipeline, in this sense, consists of all of our potential Sales Opportunities that we have in the works. Our total pipeline is the total dollar amount we could be taking in if we win every opportunity we are working on. However, most months we win on average 4-7 percent of our pipeline.

I took a sample of our opportunity data to use in this project. This data set has 13 columns, though not all were used in the project. The columns are as follows:

- PipelineRange: depending on the amount the opportunity is set at, it is put into one of 5 different range categories
- CurrentOppAmount: the current exact amount that the salesperson has set the opportunity at, how much money we expect to make from this potential contract
- IsWon: this is the metric we are attempting to predict, if the opportunity has been won or lost
- QualifiedAmount: this is the amount that the opportunity was set at when it was put into the qualified stage, which is the stage where it gets added to our pipeline

- OwnerTitle: the title of the salesperson
- SalesOffice: the office of the salesperson working on the opportunity
- SalesRegion: the region that the salesperson is located in
- NumberOfEmployees: the number of employees that the company we are pursuing has
- AccountScore: we give each opportunity an account score based off a number of metrics that the Sales team decided on
- IsClosed: if the opportunity has closed or not
- DaysPushed: if the opportunity's close date has been pushed into the future, how many days it was pushed
- MonthsOpen: how long the opportunity has been open
- OppClosedDate: when the opportunity is expected to close

This data set was then used to create an ML model to predict whether an opportunity will be won or lost.

1.2 Motivation

As mentioned in Section 1.1, at Service Express we are currently trying to update the strategies of our Sales team. Throughout the year I have worked with Service Express, I have worked closely with the Sales team in this endeavor. Over the summer I developed many dashboards to bring attention to errors and red flags within their Sales Opportunity data. A goal of our Data Science team has been to develop an ML model to predict our win rate for a given month, as well as what opportunities will be won or lost. This will allow us to give us an idea of how much money will have coming in each month. It will also give our sales people an idea of what opportunities they should focus on in a given month. My goal with this model is to give our company this clarity to become more successful.

1.3 Related work

I have lots of previous experience in this area, as mentioned in Section 1.2. Last semester, I did a project on a similar data set, looking for trends in won opportunities to see if certain metrics influenced the outcome of the opportunity. Additionally, I have spent roughly a year working closely with the Sales team to see what strategies they employ when working on opportunities. Finally, I have spent the last semester taking CMSE 202, and developed and worked on a variety of ML models for this class.

Chapter 2

Methodology

2.1 Initial Research

Beginning the project, I was not certain about what type of model I wanted to use. In class, we had looked at a variety of ML models, from linear regression to Perceptron models. However, given the type of data I was looking at and the output I was hoping for, none of these models would work. However, I knew there were many model types out there, and proceeded to research the various types of models I had access to. This led me to decided to use a Logistic Regression model, as I was using numerous metrics to predict a binary output.

2.2 Packages

I used a variety of packages in my code. I used:

1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. sklearn
    - LogisticRegression
    - train test split
    - SMOTE
    - RFE
    - confusion matrix
6. stastmodels.api

2.3 Data Input and Cleaning

The first thing I had to do with the data was load it and go through the process of cleaning it. I used pandas to load in the data. When I loaded the data, I had multiple categorical variables. With the model I wanted to use, I knew I was going to want all quantitative variables.

I had to put in a lot of work to clean the data. To begin with, I did have some nan values in the columns, so I filled those with the median value for those columns. Next, I got rid of the Industry column, as I knew from previous work that the industry had very little effect on if an opportunity was won or lost. I then created new columns for the Pipeline Range buckets, as well as the Sales region buckets, assigning a binary variable depending on if the opportunity was part of that bucket or not. Additionally I dropped the Sales Office and Owner Title columns, as I had no easy way of making those quantitative variables, and had knowledge that they had little affect on an opportunities outcome. Finally, I grabbed just the opportunities that were already closed.

2.4 Creating the Model

I first began by splitting the data into training and testing sets as shown in Figure 2.3. Once I had my training data set, I created a logistic regression model. I then found the 22 metrics that had the most influence on the model, as shown in Figure 2.1. Once I knew what these metrics were, I was able to pull them out and focus just on those metrics. I created subsets of the data with just those columns. After that I ran that subset through the logistic model. I looked at the fit and statistics of that model to see the accuracy of that model. I compared this model to models created with fewer or more metrics (a range of 19-25 metrics) to see how I could get the best accuracy. I settled on 22 giving me the best accuracy. This code is shown in figure Figure 2.2

2.5 Figures

```
data_final_vars = df_closed.columns.values.tolist()
y = ['IsWon']
x=[i for i in data_final_vars if i not in y]

logreg = LogisticRegression()

rfe = RFE(logreg,22)
rfe = rfe.fit(os_data_x,os_data_y.values.ravel())

ranking = (rfe.ranking_)
```

Figure 2.1: Creating the model

```
logit_model = sm.Logit(y,x)
result = logit_model.fit()
print(result.summary2())

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.3,
    random_state=0)
logreg = LogisticRegression()
logreg.fit(x_train,y_train)

y_pred = logreg.predict(x_test)
print('Accuracy of it is: {:.2f}'.format(logreg.score(x_test,y_test)))
```

Figure 2.2: Updating model

```
os = SMOTE(random_state=0)
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.3,random_state
    =0)
columns = x_train.columns
```

Figure 2.3: Splitting into training/testing

Chapter 3

Results and Analysis

3.1 Accuracy of Model

Overall, I got an accuracy of 82 percent. I was pretty happy with this accuracy, as previous models we have attempted to create as a company have been around 75 percent accurate. I did develop a confusion matrix to look more into the accuracy, this is shown in Table 3.1. This shows we have a pretty successful model. Additionally, I looked at the precision, recall, and f1-scores to see how the model did, as shown in Table 3.2. Finally, I also ran code to develop an ROC curve to look at the accuracy of the model, shown in Figure 3.1.

These Analysis showed me that I had a pretty accurate model to work off of.

3.2 Figures and Tables

Confusion	0	1
0	2052	414
1	484	1955

Table 3.1: Confusion Matrix

Statistics	Precision	Recall	F1-Score
0	.81	.83	.82
1	.83	.80	.81

Table 3.2: Statistics on Model

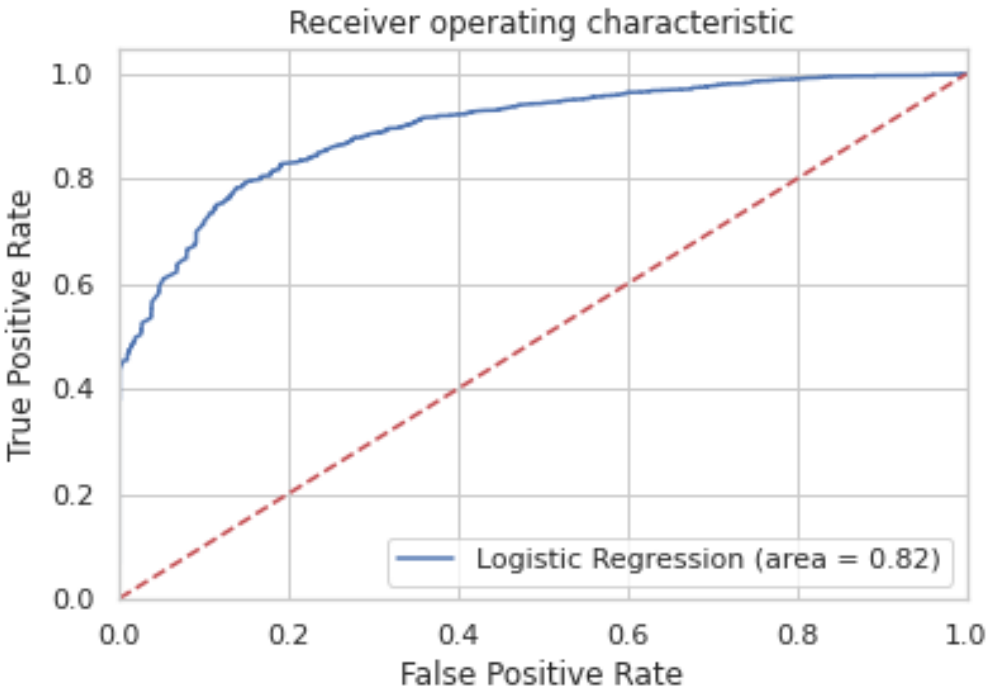

Figure 3.1: ROC Graph

Chapter 4

Conclusion

Overall, I was happy with how my model turned out. I was able to build a Logistic Regression model with 82 percent accuracy. Not only this, but I was learned about many new packages and strategies in building an ML model. I believe that this model has the ability to be implemented in my job and potentially make a big impact in the company.

4.1 Future Work

From here, I hope to work with members of my team to potentially increase the accuracy of this model. That may include adjusting the metrics I am looking at, or even looking at new metrics. Additionally, I hope to implement this model at work. If done, the Sales team would be able to use this model to see what opportunities are predicted to be won, and we can predict what of our pipeline will be won.

Bibliography

[1] Overleaf: LaTeX editor
`https://www.overleaf.com/project`

[2] Susan Li. *https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8*

[3] Data provided by Service Express