Capstone Project: (Company withheld) Win Predictability

Sujay Chebbi, Dawson Cook, Callie Gilmore, Sitong Li and Justin Wagers

Business Intelligence Capstone

McCombs School of Business, University of Texas at Austin

Capstone Project: (Company withheld) Win Predictability

Executive Overview

The objective of this project is to create a machine learning model that accurately predicts the probability that (Company withheld) will win incoming custom deals based on their characteristics. The successful creation of this model would allow (Company withheld) to better allocate their time and resources in order to prioritize deals that have a high likelihood of winning as well as a high potential for revenue. This project is focused on answering the questions related to deal prioritization such as identifying important characteristics for winning a deal and how they affect overall win probability. By answering these questions, (Company withheld) is able to prioritize these deals, which would create tremendous value and cost savings for the company. It would allow the team to focus their time on winnable deals, which will result in increased revenue and less time and costs spent on deals that are not as likely to be won. We analyzed data from (Company withheld)'s MyDeal system, which catalogs deal characteristics at a later stage in the deal lifecycle, where we have the most information about the deals. Ultimately, the team was able to create multiple machine learning models that predict a deal's win probability based on its characteristics with a satisfying accuracy and pinpoint important characteristics for prediction. We created the models using MyDeal data, tested them on a validation set and then performed blind testing on new deals coming in. Going forward, we recommend that the (Company withheld) team implements the model into the MyDeal system so that when user's input their deal characteristics, they are able to see the win probability rate. Based on this information, user's could compare the win probability of all the deal's that they are assigned and only focus on those with higher probabilities. Hopefully this will allow revenue increases and cost savings for the business.

Articulating business outcomes

Business context

Background. For this project, we are working with the Global Solutioning Group within (Company withheld). Global Solutioning works with the Sales Department within (Company withheld) to provide additional support for the sales process. The process begins by a request being sent by a salesperson to Global Solutioning containing the details of the sale including technologies and services involved in the sale. This request is sent using Salesforce data and sent through an internal system called

Custom Solutions Engagement (CSE). Oftentimes, these requests are for custom groupings of technologies or custom services. From this request, the solutioning group creates a deliverable of the offer and how it can be delivered to the customer. This deliverable is given to sales through a system called MyDeal and is reviewed and eventually will lead to the opportunity either won or lost based on whether or not the client purchases the deal. For our project, we will be working with the request and deliverable data from MyDeal to determine the likelihood that a deal will be won based on the attributes involved in the sale.

Justification and Benefit. The idea for this project is to build a model that accurately predicts the probability that a deal will be won by evaluating the components of the deal. There are many different attributes involved in the deal so we are aiming to determine which of these are most important and then predict the probability. This is important because the resulting model will allow Solution Architects to more appropriately spend their time on the deals that they are likely to win, which will ultimately generate more revenue and less wasted time on deals that they aren't likely to win. Additionally, this will allow Solution Architects to better prioritize deals to work on, which will allow them to get the deliverables to the sales division faster, thus speeding up the sales lifecycle. It is important that we also determine the probability of winning the deal at every stage in the deal lifecycle and to catalog how that changes over the entire sales lifecycle.

Business questions

Business Problem/ Opportunity that Motivated Project. The team that we are working with is a part of the Global Solutioning team so they described to us the importance of Solution Architect's role and how it affects the overall success of the sales team in regards to the sales lifecycle. There are many deals coming through the lifecycle, and often the solution architects are unable get to every deal in a timely fashion, which is a major issue. The team believes that the solutioning process can operate better to prioritize the correct deals and analytics can be used to do this. Using analytics, we can better streamline the process to ensure that the Solution Architects spend their time on deals that are likely to win and generate additional revenue. As a result, (Company withheld) will not lose out on revenue on deals they would otherwise win because the architects will be able to prioritize these, rather than spending time on those deals that aren't likely to be won. This will result in better resource allocation

for (Company withheld) and key insights about what makes a deal successful, which can be shared with the sales team.

Specific Business Questions. The main and most important business question is what is the probability that a deal will be won based on the attributes and information given in the request form. We also need to determine what attributes and information in the request form are the most important as far as having predictive value but also in determining successful deals. We would like to gain insights into what attributes make a deal more likely to be won. For instance, we want to understand trends in different regions, technology types, deal complexity that ultimately make a deal more successful.

How the Business Questions Are Related. All of the business questions are centered around understanding the probability of winning the deal. This is our ultimate objective and the main focus of the project. The other business questions are stemming from understanding more about what goes into determining the probability and understanding key insights that can be gained from our model. Our goal is to focus on building a successful Machine Learning model that can predict the win rate accurately but also to gain key insights by diving deep into the data, variables and their roles and relationships within the model.

Business outcomes

Business Outcomes Expected by Client. The client should expect a classification model that can predict the probability of a deal being won. With the probabilities of a deal available to (Company withheld), they can carefully choose which deals to pursue, resulting in a more cost efficient business strategy. Additionally, they expect to obtain key insights about what attributes are important in winning a deal as well as the attributes relationships to each other and overall trends that can be derived from the data.

Business Outcomes and Strategy Connection. Knowing the information about specific deal probabilities, (Company withheld) can strategically pick and choose which projects to pour more resources into. As a technology company, (Company withheld) has an offensive strategy in order to gain new sales and revenue. This project aligns with an offensive strategy in order to prioritize deals and to aggressively go after deals that they are likely to win. The business outcomes of this project will result in (Company withheld) being able to better position themselves to compete and win sales.

Expected Business Value or ROI. The potential business value of this project is to prioritize deals that are likely to be won. The result is that (Company withheld) will not be losing as many deals that can be won. This turns lost revenue into generated revenue and better allocates (Company withheld)'s resources. This lowers the overhead cost that (Company withheld) is spending on each deal and brings in revenue from deals that would not have been prioritized and lost because of that. The business value of this project is very exciting and can potentially have a major impact on the solutioning and sales processes.

Exploring source data

Available Data

We received data from the MyDeal system. MyDeal pulls data from a form that is filled out by the Sales Team for each deal when they have a new deal in the pipeline that requires additional solutioning from the Global Solutioning Team. Our raw dataset consists of 114,641 rows of deal data with 104 columns. For the most part, there is only one deal per row; however, there are a few instances where one deal continues on more than one row where there is additional information regarding the deal.

Business Description of Data

The data consists of deal information and contains 104 different columns. The type of information includes Region, Country, Forecasted Revenue, Estimated Cost, Contract Margin, Solution Architect (encrypted), Customer Name (encrypted), Technology Name, Deal Status, many different date values from the deal lifecycle and many other fields. Overall, the dataset contains all the deal information necessary for the solutioning team to solution the deal and provide the deliverable to the sales team.

Data Limitations/ Issues

Due to the manually input nature of some fields, the data required a lot of cleanup. There are many null values on necessary columns, while some columns are almost entirely null. Additionally, the (Company withheld) team informed us that certain columns, such as the cost, margin and revenue

columns, should be scrutinized carefully, as often the sales team will inflate the numbers or simply put 0 if they do not know. There are a large amount of categorical variables where there are many different category options, such as the technology and country fields.

Data Transformation/ Preprocessing

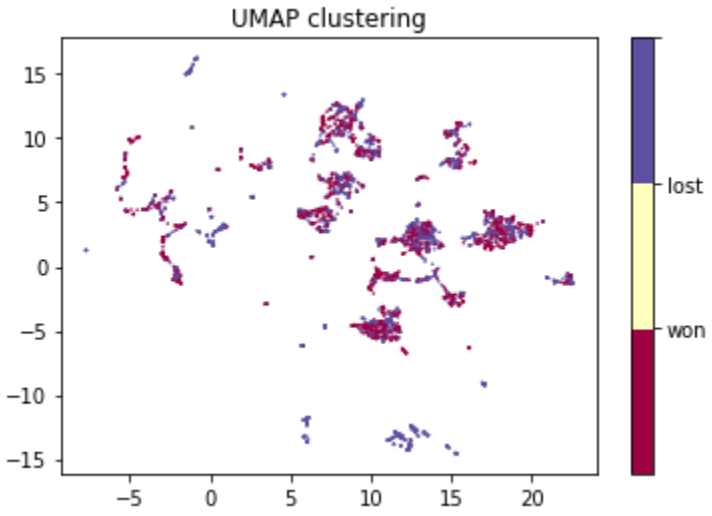
Below are some of the many of the steps we took to prepare the data:

- Eliminated variables with more than 50% null values
- Transformed our dependent variable (deal status) to a binary variable representing win or loss, eliminating deals that currently are in progress
- Transformed the date variables by converting them to match (Company withheld)'s fiscal calendar and then using Quarters and Years
- Calculated a "Time to Solution " variable by subtracting the intake date of a given deal from its final delivery date
- Consolidated deals with multiple rows/technologies into the format of one row per deal, adding complexity metrics as a new column in the case of these compound deals
- Conducted basic mode imputation for variables where null values should be represented by zeros: Source System, Number of SOW Revisions

Feature Extraction and Dimensionality Reduction

The MyDeal dataset contains many categorical variables, some with up to 200 levels. Performing one-hot encoding for this set of predictors would give us a very sparse matrix, which is undesirable for modeling and computational intensity purposes. Thus, we had to figure out an approach to reduce the dimensionality of this data.

For this dimensionality reduction, we considered three different approaches: binning, PCA, and UMAP (Uniform manifold approximation and projection). Among the considerations in deciding between these approaches were their validation accuracy in deal classification, ease of interpretability, and ease of transformation with new data. Cross validation was performed for each of these approaches to determine the optimal number of bins for each variable in the binning method, optimal number of components in principal component analysis, and optimal number of dimensions for UMAP. Below we can see a two-dimensional representation of the entire training set using UMAP.



Despite our high hopes for UMAP as a relatively newly developed advanced dimensionality reduction technique and PCA as a tried-and-true method, we were finding that these strategies were only marginally increasing the performance of our classifier model. Manually binning some of our variables, however, allowed us to have greater creative authority over the structure of the data, combining variables that we suspected to be related through our own insights and guidance from (Company withheld). As a benchmark, our validation accuracy was jumping from 64% to 68% in a simple logistic regression model after this binning. Below, we detail how binning was done.

The critical variable Technology Name had over 140 levels, so we decided to bin the different technologies into complexity groupings based on the mean time to solution*(Figure 5)*. This reduced the variable to 5 levels (optimal number of bins chosen through cross validation), from lowest time to solution technologies to highest time to solution technologies. Our hope was that this would be a reasonable proxy for the technology name itself while at the same time reducing complexity. The same approach was taken for the Opportunity Type variable.

Similarly, we sought to reduce the dimension of the Country variable from nearly 200 down to 5. For this, we broke the variable into two components: the region (one of four major business regions for (Company withheld)) and the volume of deals done within that country, broken down into 5 bins from countries with lowest to highest volume of dealings with (Company withheld)*(Figure 6)*.

Our pricing variables included Contract Cost, Margin, and Forecasted Revenue. Cost and margin were relatively reliable fields and only required minimal mean imputation to increase usability. In the

case of Contract costs, we decided to conduct quantile binning as a method of avoiding weights on outliers*(Figure 7)*. Forecasted Revenue was much trickier, as it is simply a human estimate of the revenue for a given contract. However, it was showing up at the top of feature importance tests and we still wanted to use it in our model, so we decided to conduct median imputation.

Perhaps most importantly, we sought to extract customer characteristics from the data. We had access to customer ID, but had to dig deeper to find metrics to make this predictor more useful. For this, we created a customer complexity metric that placed customers into various bins based on their mean time-to-solution of a deal *(Figure 8)*, and also tried to take customer history into account by measuring the number of deals a customer had done with (Company withheld) over the span of the training data *(Figure 9)*.

Analyses (Modeling, simulation and optimization)

Variable Selection

After feature extraction and dimensionality reduction, we ran a Random Forest feature importance to determine which predictors would be useful and which were less important. The results led us to use the following predictors for our classifier models:

Customer (complexity)	Region Value	Fiscal Quarter
Contract Cost	Contract Margin	Deliverable Type
Opportunity Type (complexity)	Governance Status	Is Federal
Source System	Deal Type	Has Inner Technologies
Number Of SOW Revisions	Multiple Row Complexity	Technology (complexity)
Country (deal volume)	Customer (count)	

We can note that within this set we have a mix of continuous, binary, categorical, and ordinal data types. For modeling, the categorical variables are one-hot encoded, while the processing of the ordinal variables depends on the model.

Model Training (Preparation for Blind Test)

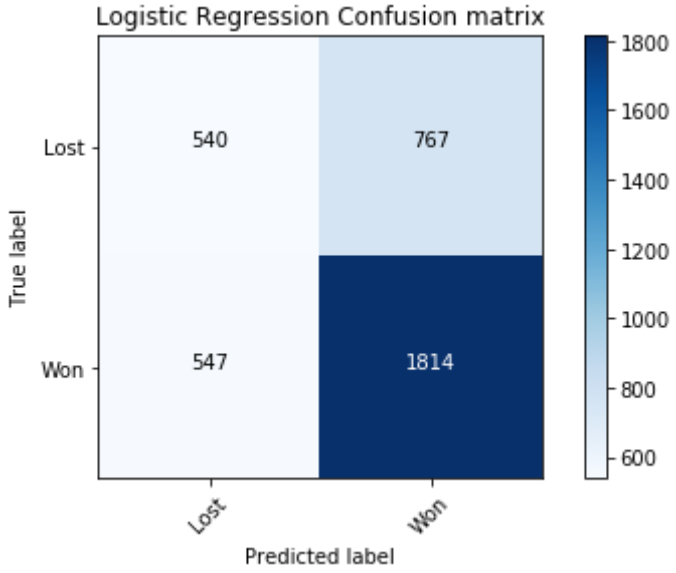
With our objective of accurately predicting deal win probability in mind, we evaluated a set of classifier models and their predictive power on holdout data. For our initial model selection, we trained the model on the first three quarters of 2020 and predicted deals in the last quarter of 2020.

Logistic Regression

The tried-and-true logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. This means that the logistic regression model has a certain fixed number of parameters that depend on the number of input features, and they output categorical predictions.

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency while requiring low amounts of computation power. Resultant weights found after training of the logistic regression model, are found to be highly interpretable. It is also less prone to over-fitting in comparison to other classifiers. For these reasons, we decided to apply logistic regression as our baseline.

After properly fitting the logistic regression classifier including parameter tuning, our predictions on the test set are as follows:



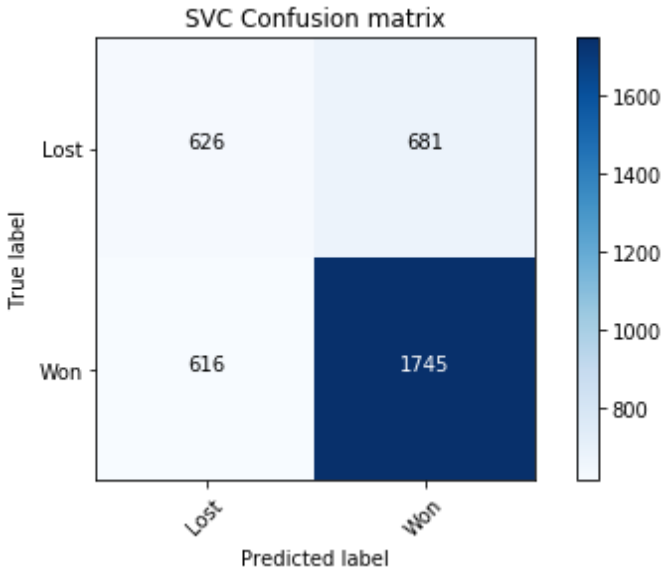
Our predictions on the fourth quarter of 2020 had a 64% accuracy, with most of the errors falling in the false positive category. (Company withheld) has expressed that these types of errors are preferable as they would rather pursue a deal and have it be lost than not pursue a deal that would have been won. Overall, logistic regression performs well as a baseline, although it leaves room for improvement with cutting down false negatives.

SVM

A support vector machine is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, it's able to categorize new data. A support vector machine takes these data points and outputs the hyperplane that best separates the tags. The hyperplane is chosen by maximizing the margins from both tags, or whose distance to the nearest element of each tag is the largest.

SVM is more effective in high dimensional spaces, working well with our data since we had to create dummy variables for all of our categorical type variables(Because SVM does not accept ordinal data). SVM also has L2 Regularization feature. So, it has good generalization capabilities which prevent it from over-fitting.

After properly fitting the support vector machine classifier including parameter tuning, our predictions on the test set are as follows:



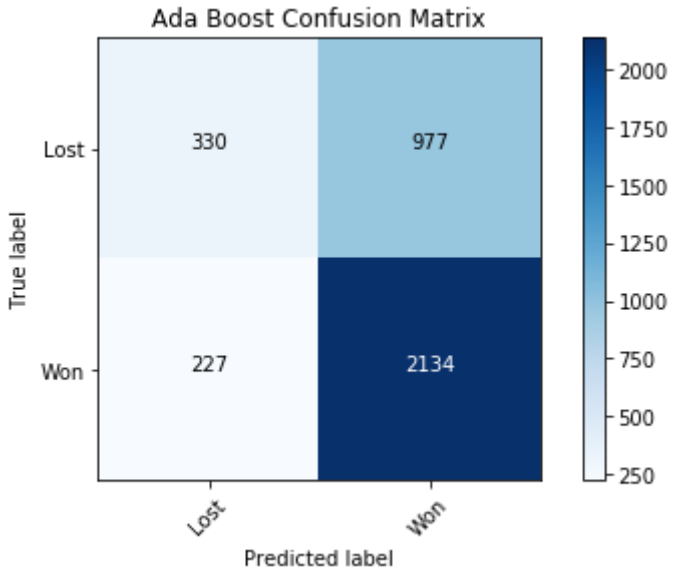
Our predictions on the fourth quarter of 2020 had a 65% accuracy, with most of the errors falling in the false positive category. The SVM model performs better than our baseline, and provides a good start to the next step in our process.

ADA Boost

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations.

AdaBoost is easier to use with less need for tweaking parameters unlike algorithms like SVM. Theoretically, AdaBoost is not prone to overfitting, so is a great choice when dimensionality is a decent size. Another great trait of AdaBoost, is that it accepts ordinal data which eliminates a step in our data handling process.

After properly fitting the AdaBoost classifier including parameter tuning, our predictions on the test set are as follows:



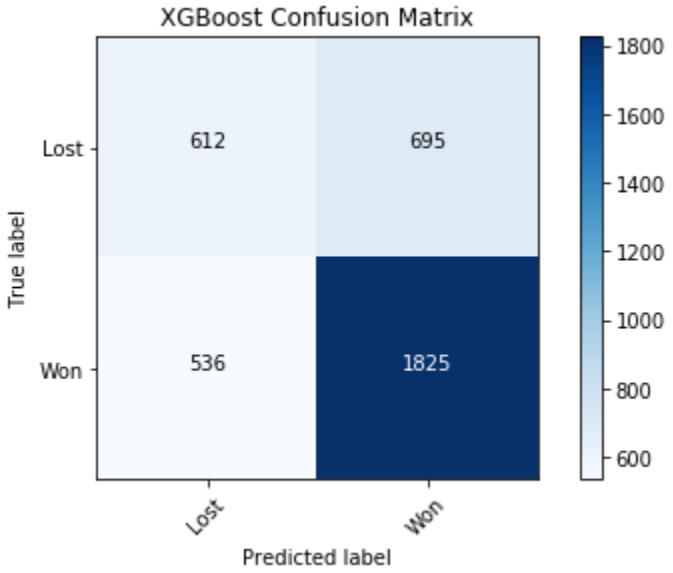
Our predictions on the fourth quarter of 2020 had a 67% accuracy, with most of the errors falling in the false positive category. AdaBoost performed very well, and had the lowest false negative rate of all of the models we tested. However, as we will discuss below, the false negative rate did not necessarily carry through to the blind test.

XGBoost Classifier

XGBoost is a tree boosting algorithm that minimizes a regularized objective function that includes convex loss and a model complexity penalty. In essence, it adds decision trees sequentially to correct the errors of previous trees until no further improvements to the model's error can be made. XGboost is a popular algorithm because of its speed and generally high, which is an important feature for ease of implementation at (Company withheld).

XGBoost and other decision tree algorithms have some major advantages over models such as Logistic Regression. For example, decision trees naturally handle multicollinearity, do not require normalization of inputs, and can perform clean handling of ordinal data, which are all quite useful in our case.

After properly fitting the XGBoost classifier including parameter tuning, our predictions on the test set are as follows:

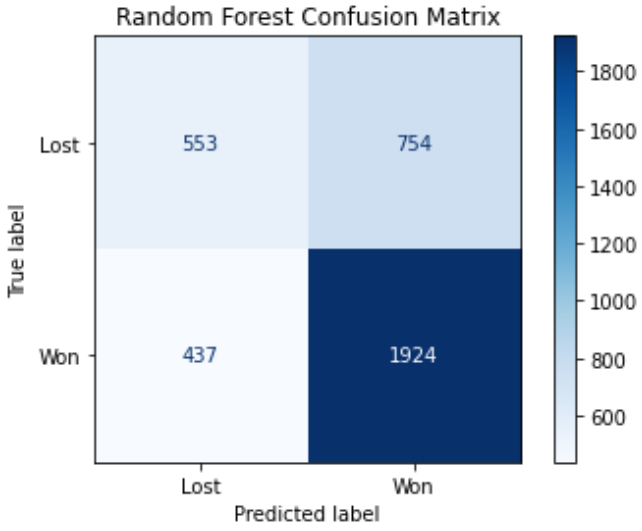


Our predictions on the fourth quarter of 2020 had a 68% accuracy, with most of the errors falling in the false positive category. Overall, the XGBoost performs quite well compared to the other classifiers, although it leaves room for improvement with cutting down false negatives.

Random Forest

Random Forest is an ensemble classifier that aggregates the predictions of many decision trees that predict poorly on their own into a single model. The combination of these weaker trees reduces the variance of prediction errors and improves upon the accuracy of the individual models themselves.

Random Forest classifiers are generally quite fast once parameters are chosen, and we saw that in our model run times. They are also known for being quite accurate, and we suspect an ensemble model might perform more robustly on blind test data. One disadvantage of Random Forest is the time intensity of parameter selection. Although the model was efficient once fitted, our selection of max depth, minimum samples, number of estimators, and maximum features took quite some time.



Our predictions on the fourth quarter of 2020 had a 68% accuracy, with most of the errors falling in the false positive category. Overall, the Random Forest performs on par with the XGBoost classifier, albeit with slightly false negatives.

Blind Test Results

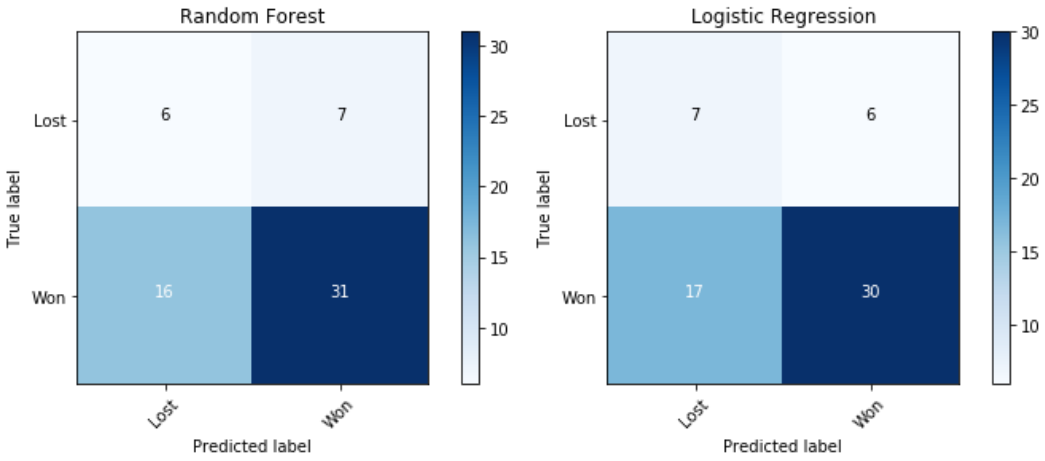
One of the most important characteristics of any given classifier to (Company withheld) is how it performs on new data. Although our models had been tested on fourth quarter 2020 data, it was important to evaluate them on the most current deals.

Thus, (Company withheld) provided us a list of 60 of the most recently concluded (won or lost) deals to evaluate the robustness of our models. We trained each model on the entire set of (Company withheld) ‘s deals from January 2020 through March 2021. The accuracies on this blind test data are shown below.

Model	**Random Forest**	**Logistic Regression**	**SVM**	**ADABoost**	**XGBoost**
Accuracy	61.5%	61.5%	53%	53%	50%

Upon initial glimpse, we notice two things. First, the models do not live up to their projected accuracies from predicting on exclusively 2020 data. For example, ADAboost had a 67% accuracy on fourth quarter 2020 data, but a significant drop to 53% on the blind test. Second, We see that Random Forest and Logistic regression both performed quite well in the blind test. After further investigation, it was noted that the blind test consisted of 78% won deals. We believe Logistic Regression was so successful in this case because of its bias towards predicting a deal will be won.

To compare the results for the two best classifiers, we can examine their confusion matrices.



This comparison shows only one error difference between the models with Random Forest making one fewer false negative error than logistic regression. Both models make significantly more false negative than false positive errors, although both perform well in comparison to our other classifiers tested.

both the robustness of an ensemble classifier (Random Forest) and the general reliability of Logistic Regression.

Final Model Selection

Our results speak to the difficulty of model performance in real-world blind tests. The business environment for (Company withheld) is constantly evolving, but we are using historical data to predict the future. However, 61.5% accuracy, as confirmed by our (Company withheld) counterparts, is a fantastic result on blind data. We believe that because of the ensemble nature of the random forest classifier and low false negative rate compared to our other models, this model will continue to succeed

in predicting deal outcomes for (Company withheld) as it is continually trained on new data. This model in the hands of ground-level solution architects will lead to efficiency improvements in the deal lifecycle.

(Company withheld)'s grand objective is to maximize profit. Solution architects play into this profit maximizing equation by spending their time on deals that are more likely to be profitable for (Company withheld). At the current moment, these architects can hardly determine whether a deal will be won or lost better than a coin flip (the average win rate is just over 50%). Our new classifier model can push that coin flip 50% certainty all the way to 62%, likely catching deal characteristics that solution architects accidentally overlook or aren't able to extrapolate from the surface level data. This leads to greater time efficiency for solution architects at (Company withheld) as they use the win probability as a metric to determine which deals are more deserving of their time.

Recommendations, Operational Execution and Change Management

The team sees the installation of our model to be two-fold. First, the model output can be incorporated into the MyDeal platform for solution architects to actively see a win percentage for each deal they manage. Second, a scenario exploration tool will be provided to solution architects to evaluate a given set of deal characteristics versus win percentage for those characteristics.

For the MyDeal output, the (Company withheld) custom solutions team is tasked with the logistical challenge of providing a real-time model output for deals as they are initiated and updated. This will involve collaboration between the IT Architecture and Custom Solutions Sales teams. As of now, IT Architects have already begun integrating the model as an Excel-based macro tool that will provide a probability output based on the deals in the spreadsheet. Solution architects will soon be able to select a subset of deals they are interested in and see the win probabilities for those deals. The model itself will be pre-trained on the past 18 months of data (barring any major external events, mergers, etc.) and re-trained quarterly at minimum to ensure current and accurate results. In the long run, we have discussed with the IT Architecture team the possibility of having a constantly updating model output; this would make it so every time a new deal is added or updated, probabilities are adjusted accordingly. The ease of access to these win probabilities is essential in maximizing the time solution architects spend on revenue-producing deals, and the implementation of the model has already begun.

The scenario exploration tool would operate slightly differently, allowing solution architects to select and change deal characteristics and see the resulting win probability. While we have created the basic infrastructure for this tool, the ideal implementation would involve a user-friendly interface for ease of exploration. This will again be a collaborative effort between the IT Architecture and Custom Sales teams, and user-friendly implementation may be a few months out. The scenario exploration tool would be especially useful for evaluating deals that are early on in the lifecycle, but for which the solution architect has some suspicion about its characteristics. It also provides a general intuition for these architects on which characteristics lead to higher or lower win probabilities.

In its entirety, installation of these two tools will involve three steps: development, deployment, and tuning. In the development phase, IT Architects, including the advisors for our project, will fine-tune the model and create the digital architecture required to integrate the model output into MyDeal and into a scenario exploration tool. In the deployment phase, the architecture will be put in place, and solution architects will start receiving deal win probability as a field in the MyDeal stage of the process. While this is easier said than done, a clear communication between the IT Architecture team and the Sales team will help ensure a smooth transition. This will include a brief training for solution architects by the IT Architecture team on how to interpret the win probability of a deal and how they should use this metric as they decide which deals to prioritize. Lastly, the tuning phase of installation will involve feedback from individual solution architects and sales managers on how the tool is working, if and when they find it to be useful, and how it can be improved. This feedback will be extremely valuable to the IT architecture team as they move towards a continuously updating win probability model. The teams should track utilization percentages in terms of revenue-generating projects to evaluate whether this win probability output is providing the business value it is expected to.

Conclusions

The purpose of this project was to create a machine learning model that can accurately predict the probabilities of a (Company withheld) deal outcome given certain deal characteristics. With the completion of our task, (Company withheld) would have access to win percentages of individual deals. With this value, (Company withheld) could allocate resources to deals that are more likely to be won, which will lead to better time and cost efficiency. The team did run into some minor roadblocks, as data was not properly cleaned and had some flaws regarding quality. Many of the variables contained a high

number of null values, and we also faced the issue of dimensionality reduction. With the help of imputation and feature engineering, these issues were solved to the best extent possible given the current state of the data. Our Random Forest Classifier will provide (Company withheld)'s global solutioning team great value in the future. With the adoption of our model, it will allow the custom solutioning team to have an increasingly accurate prediction on the win probabilities of deals.We expect (Company withheld) to implement our machine learning model into the MyDeal stage as soon as possible in the MyDeal stage of the deal lifecycle and into a scenario exploration tool for solution architects. We hope that our models accuracy will be continually improved upon and used earlier on in the deal lifecycle as data collection and modeling improves.

Appendix

Figures

Exploratory Data Analysis

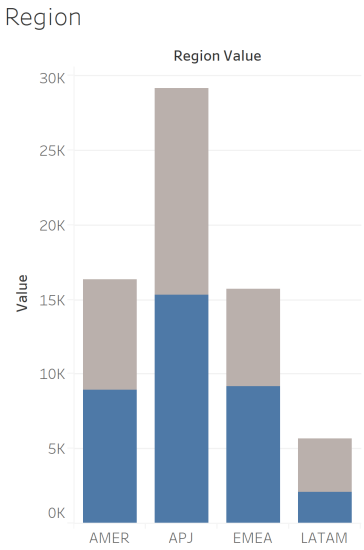


Figure 1. Wins and losses per major geographical region

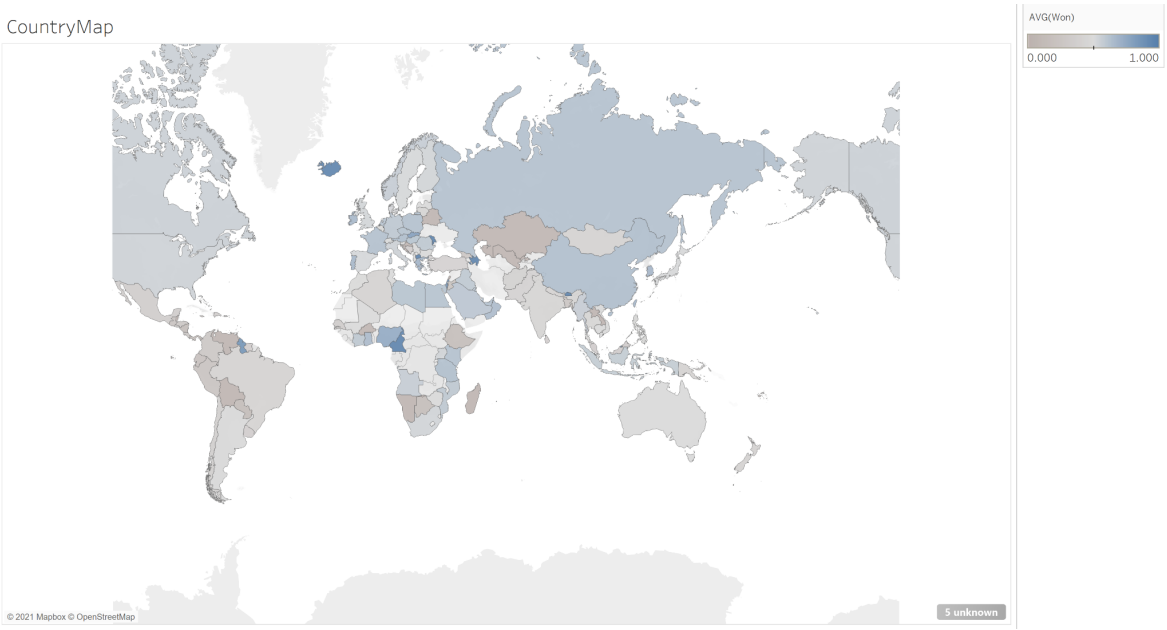


Figure 2. Gradient visualization of wins and losses per Country

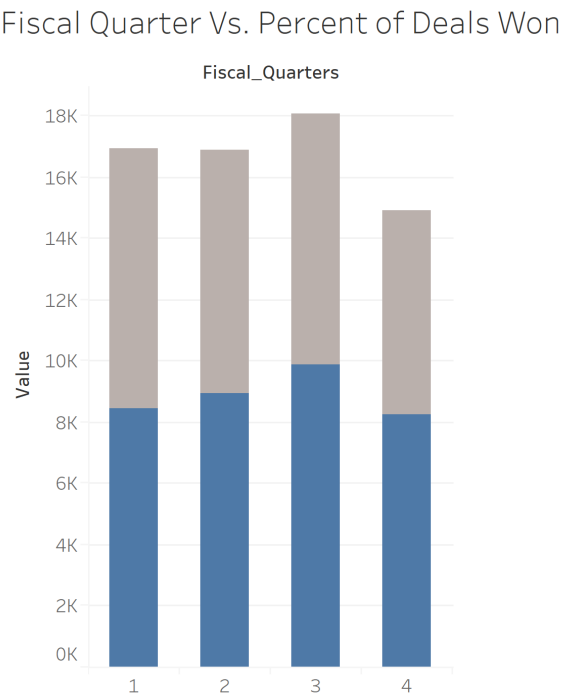


Figure 3. Wins and losses per Fiscal Quarter

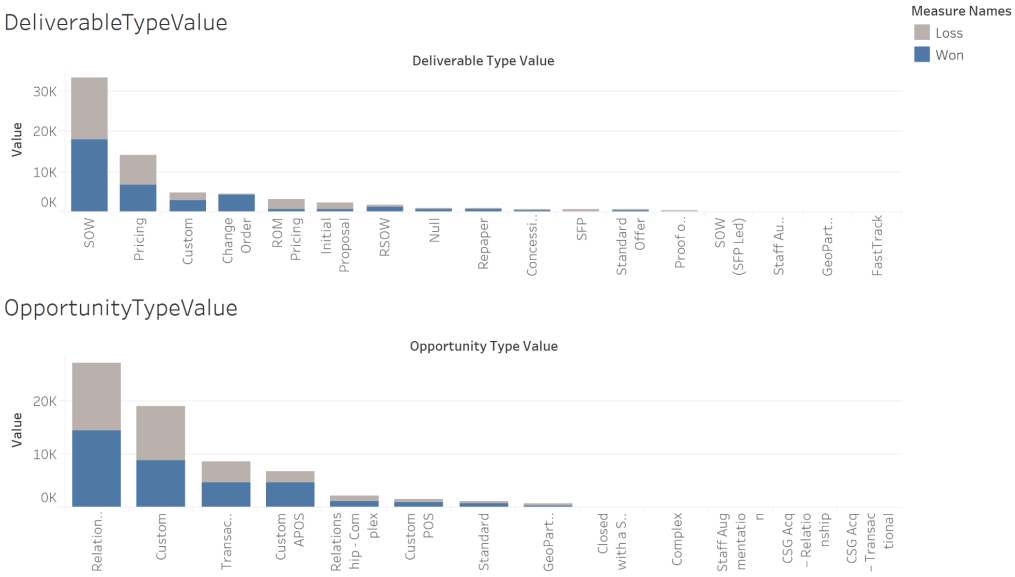


Figure 4. Wins and losses per Deliverable Type Value and Opportunity Type Value

Feature Extraction

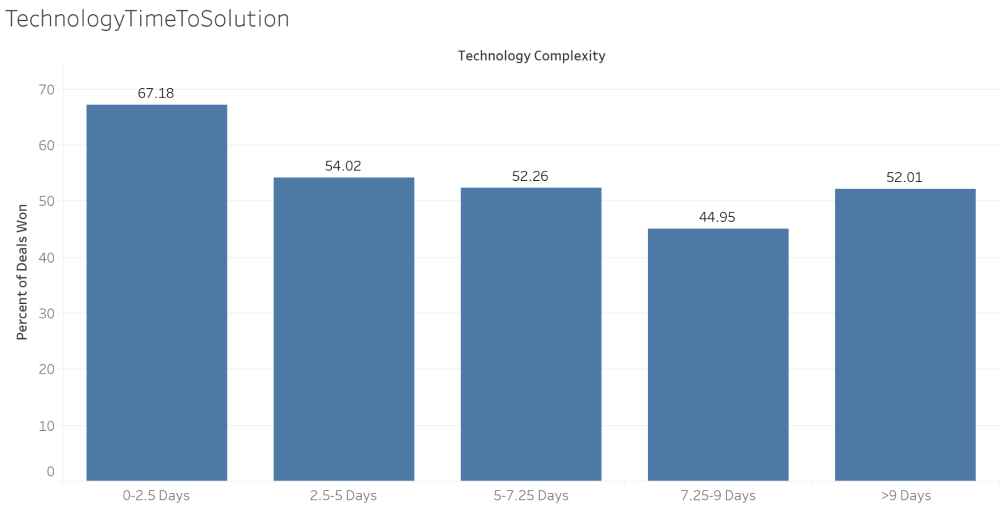


Figure 5. Binned Technology using average solutioning time

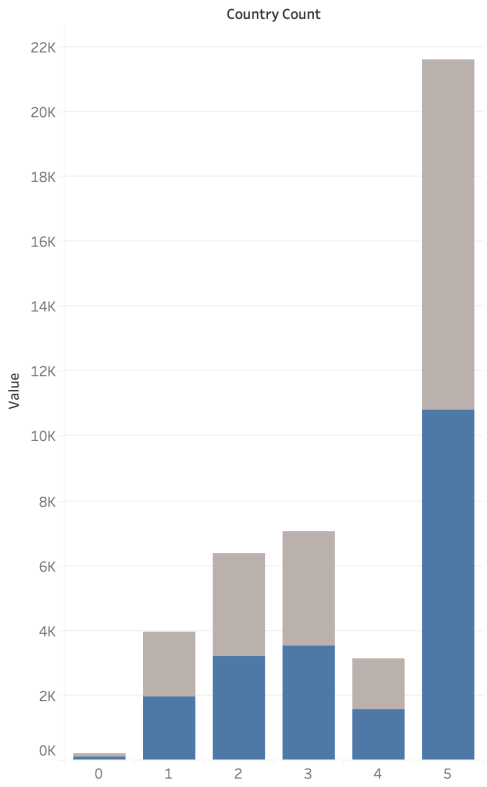


Figure 6. Binned Country using history deal counts

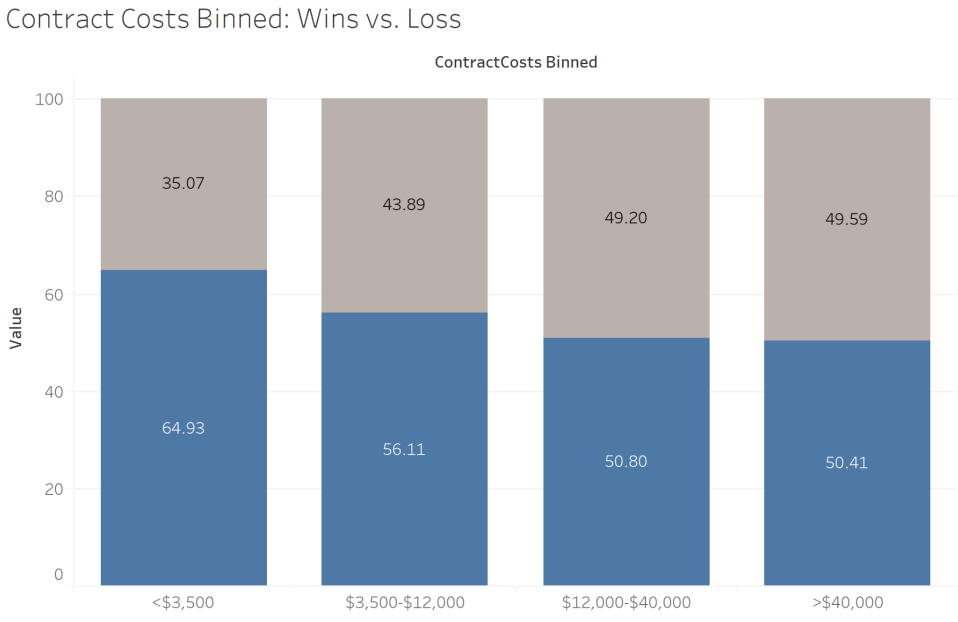


Figure 7. Binned Contract Cost

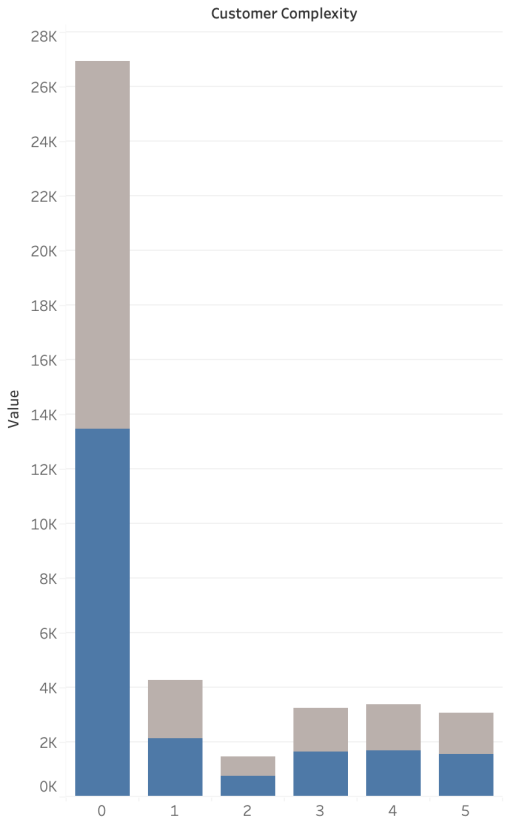


Figure 8. Binned Customer using average solutioning time

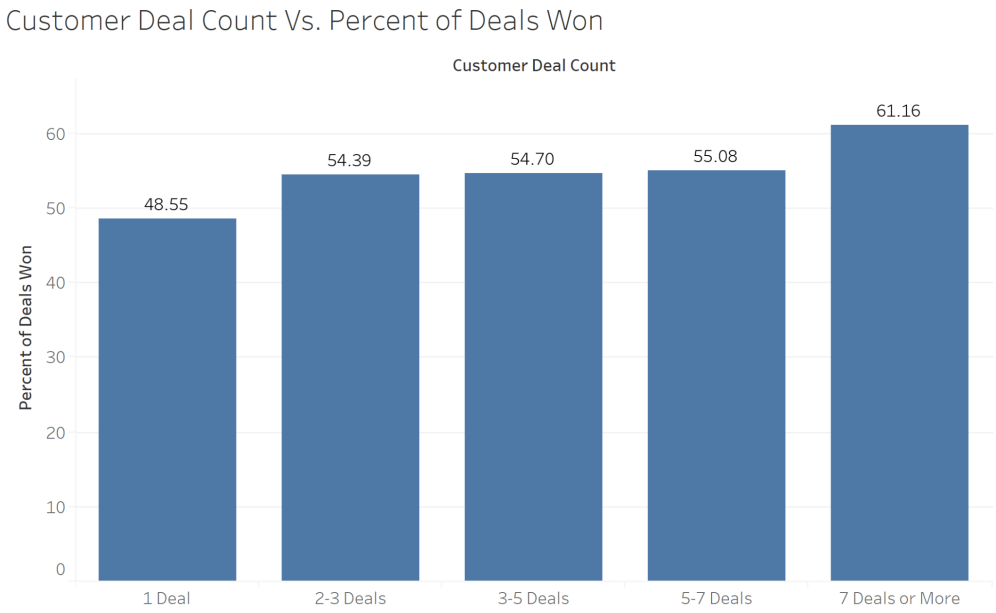


Figure 9. Binned Customer using history deal counts