Purpose

The purpose of this project is to correct inaccuracies in the Company's sales forecast. Originally, the scope of the project was to do so by using Time Series analysis techniques such as ARIMA, Holt-Winters exponential smoothing, and a 'by-hand' analysis that removed the average quarterly error from the forecast. After the initial presentation to leadership, the Company's CFO requested additional forecasting be added, using Binary Classification on the sales pipeline to predict the ending Closed Won number at a deal level. She also requested to see similarities between Closed Won deals and Closed Lost deals. This was accomplished through Logistic Regression and feature extraction in Classification Trees. While the Company updates its quarterly forecast weekly, part of the purpose is to provide the more accurate Closed Won dollar amount earlier in the quarter, so Finance and Company leadership can make decisions within a smaller margin of error. The Time Series forecast is meant to give one number that stays the same over the course of the quarter, while the Binary Classification can be updated on request. I also hope these more accurate forecasts will also pave the way for a better relationship with the board of directors and investors, who tire of seeing a large drop in forecast at the end of each quarter. They too have the right to the most realistic sales outcome.

Background Information

The current sales forecasting process at the Company is tracked primarily in weekly Excel sheets pulling data from Salesforce CRM and is based on a combination of the salesperson's intuition and their team leader's overrides, also based on industry expertise and intuition. Sales leads pay close attention to their teams and are generally aware of who on their team forecasts more conservative numbers and who tends to forecast more aggressively. From there, the leaders override either the forecasted contract value or the deal stage to compensate for the salesperson's leanings, but consistent turnover and hiring make it difficult to track long term performance. This has been the method of forecasting at least since I started at the Company three years ago.

The sales forecast is run quarterly with greater emphasis placed on the Weeks 4 and 8 forecasts. In the Fiscal Year 2024 (02/23-01/24), the average drop between the Week 4 forecast and the final Closed Won number across all quarters was 10.14%. The average drop between the Week 8 forecast and the final Closed Won number was 10.78%, meaning the forecast grew further away from the final number over the course of the second month of the quarter. The problem is obvious, Sales gut feelings cannot provide accurate sales forecasts.

Methodology

Time Series

Forecasts are done in Excel and are kept in an easily accessible drive, but regular updates to the forecast formatting and information required to compile the forecast make historical forecasts and Closed Won numbers difficult to collect by automation. I collected the Week 4, Week 8, and Closed Won forecasts by hand as far back as we had record, to about 2017. Once we started the analysis, we noticed a pattern change in the forecast numbers in 2020. Upon questioning the project sponsor, I learned the Company underwent a merger then, and threw out all data before the merger to maintain consistent conditions for the numbers.

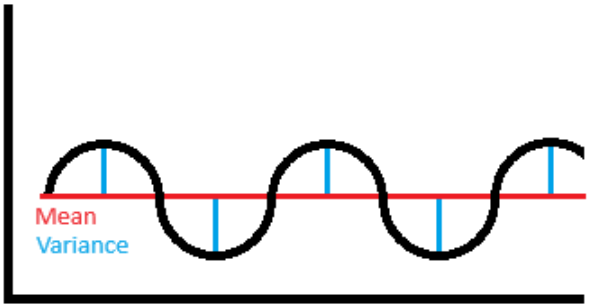
For the Time Series analysis, I wanted to give an ending Closed Won number that wouldn't change throughout the quarter. As the Company's primary forecaster, the Closed Won number is the most important number to me. We use forecasts for best case scenarios, 90%-confident-this-will-close

forecasts, and the weekly size of the pipeline and changes to the pipeline that the sales team can draw from if something falls through. I discarded all those numbers, because I wanted to predict the final Closed Won number and nothing else. Company leadership is hoodwinked every quarter by impressive best-case scenario presentations, and every quarter they are confused at the drop in forecasting in the last two to three weeks.

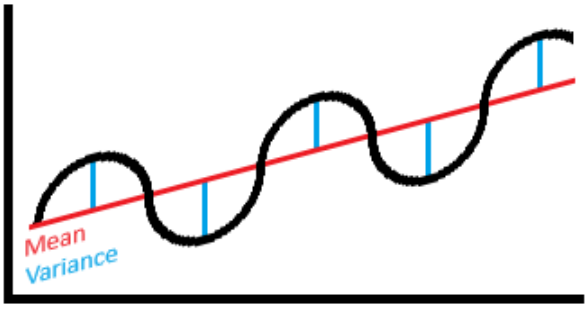
Disclaimer: I have real RMSEs for each method, but confidentiality dictates that I cannot share those errors. Instead, I have applied those errors as a measure against more recent Closed Won numbers and the RMSEs in this paper are expressed as percentages instead of dollar amounts.

I started by taking the Week 4 forecast given by the sales team, subtracting the final Closed Won number over the last several quarters, and taking the Running Mean Squared Error. The resulting RMSE ended up being a little higher than the aforementioned 10.14%, about 13.53%, meaning compared to the 2020-current training period, the forecast in the last year has improved and is closer to reality. This sales team RMSE remained the largest as Time Series methods were added.

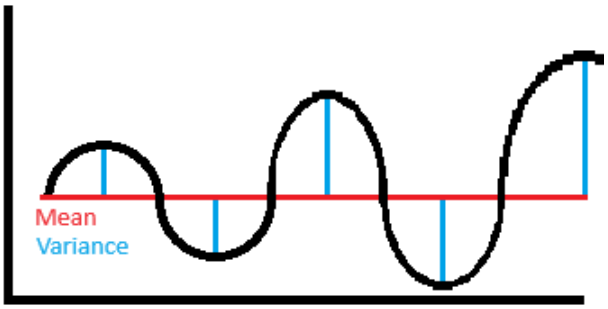
The next step was to see if the data was stationary or non-stationary over the training period. Time Series data is stationary if the mean and variance remain constant over time and is non-stationary if one of those two changes over time. These expertly drawn images I made in Paint display the difference between stationary and non-stationary Time Series data.



This data is stationary, because the mean and variance remain constant throughout the cyclical data. Examples of stationary data can include stable sales cycles, number of miles driven by a taxi driver over the course of the year, and dollars spent on groceries each month by a family of four adjusted for inflation.

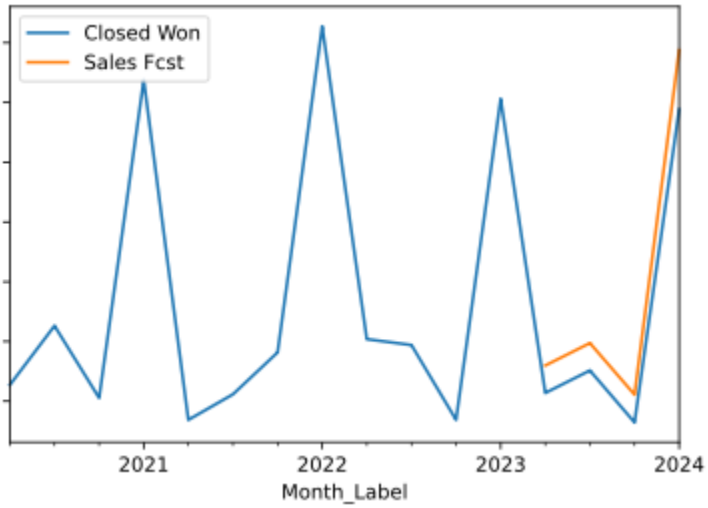


The following two graphs are non-stationary, but for different reasons. In this first one, the variance stays the same, but the mean increases as time goes on. This could also include a sales cycle, but one wherein sales are increasing over time, or the number of student enrollments in an analytics master's program growing in popularity and reputation.



Finally, this last image is non-stationary data because even though the mean is constant, the variance is changing over time. This type of data represents events that are becoming more extreme, such as wealth gaps between socio-economic classes, the strength of earthquakes around the world, or variable costs of an organization.

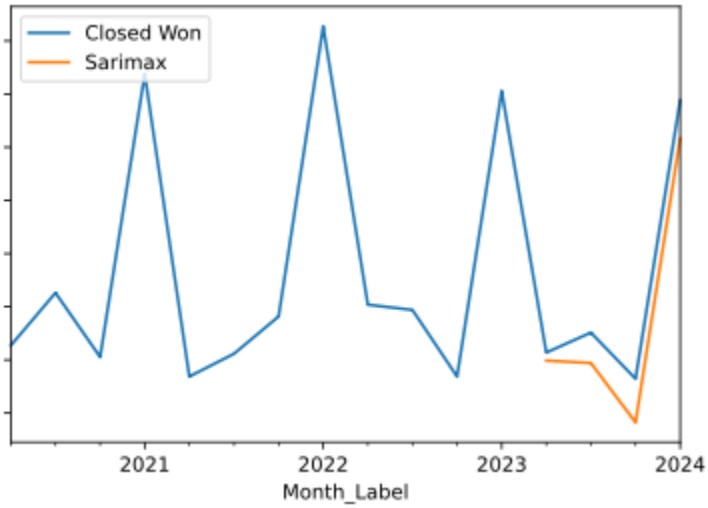
Time Series Training & Testing Visualizations



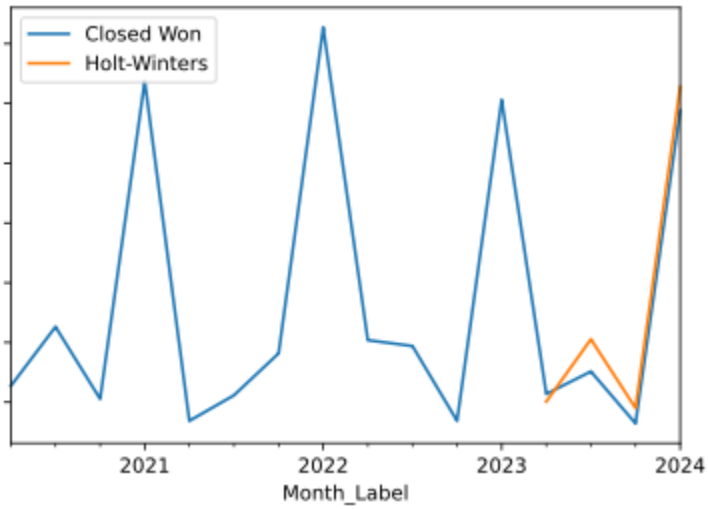
This is the Company's sales cycle starting in Q1 of 2020, with the y-axis redacted for confidentiality on the exact sales numbers, but the pattern we can use. In order to maintain consistency across testing and forecasts, I am only showing the most recent year of Sales forecast compared to the Closed Won. Each point on the orange line represents the Week 4 forecast of the Sales team. Each one is much higher than the blue line, where we finished the quarter.

Visually, it is obviously seasonal, but it is difficult to tell by sight if this data is stationary or non-stationary, so we administered the Augmented Dickey-Fuller test to find out. The null hypothesis for the ADF test is that the data is non-stationary. The resulting p-value for the test was 0.22, meaning I failed to reject the null hypothesis, so I continued the analysis under the assumption that the data is non-stationary. Non-stationary data excludes a few methods from being best practice. Methods like SARIMA (Seasonal AutoRegressive Integrated Moving Average) generally perform best on stationary data. That doesn't mean we can't use those methods and see what works best for us, it just means the results may not be as good as they could be.

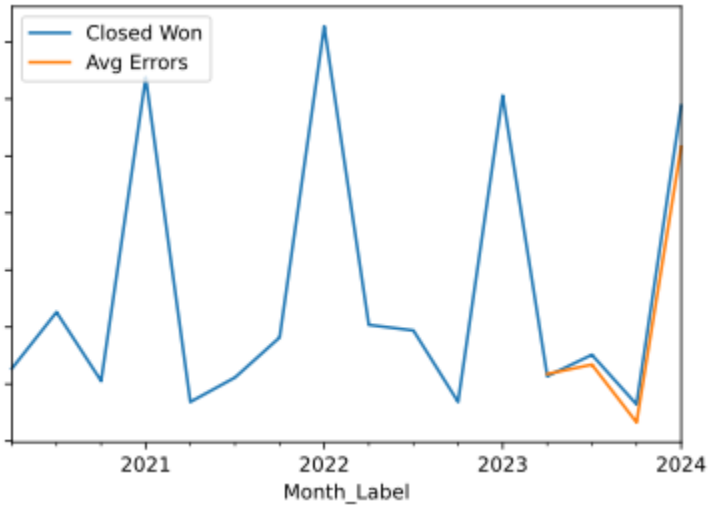
SARIMA was the first analytical method I tested, specifically for the seasonal handling ability of the method. I used the auto_arima() function to find the best p, d, and q values and seasonal order ((1,1,0) and (1,1,1,4)) and applied the forecast to the testing data. The SARIMA RMSE beat the sales team's RMSE by a low 5-digit number, about 13.15% error compared to the sales team's 13.53%. This was initially disappointing, but not that surprising since SARIMA is made for stationary data, and this is non-stationary. Eventually, disappointment turned to optimism when I realized that even a model so poorly equipped to handle this data still outperformed the daily practice.



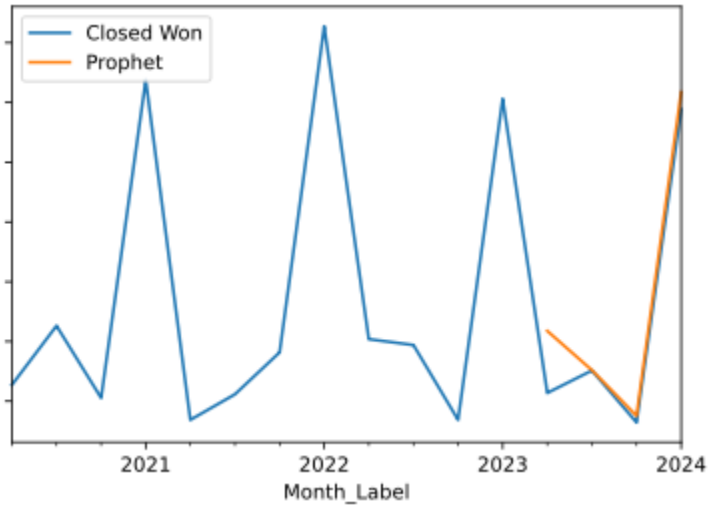
The next model was Holt-Winters. I experimented with Holt-Winters in Excel during my first semester in ISYE 6501 and had some promising results, beating the sales forecast by a sizeable percentage. After some trial and error using the ExponentialSmoothing() function, the trend was multiplicative, and the seasonality was additive. Forecasting out 4 quarters, Holt-Winters' RMSE was a 6-digit number, representing an error of 7.73% in the testing data.



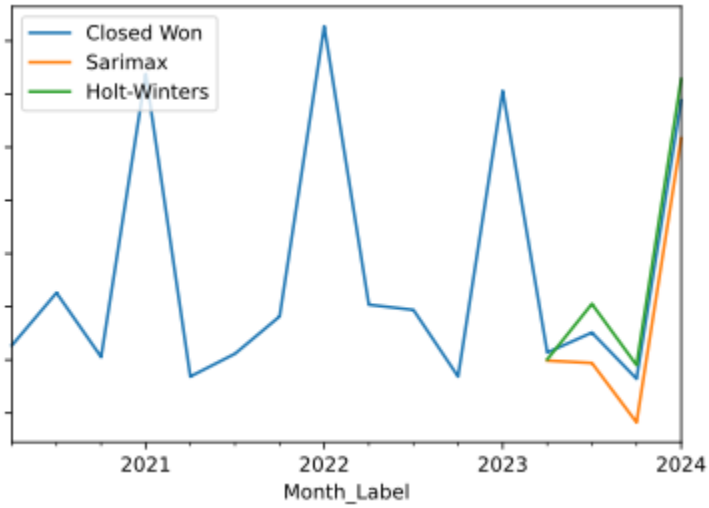
When working with Holt-Winters in Excel, I found greater success in sorting the HW forecasts by quarter, calculating the average quarterly error between that forecast and the actual Closed Won number, and then removing that average quarterly error from the calculated forecast. After doing the same in Python, I found the RMSE of my 'by-hand' error compensation was worse than the machine on its own, 8.58% error. In this analysis, I identified and removed the sales periods before the merger. In the Excel experiment, I kept them in, having not known about the merger at that time. Since the merger, sales have increased, and so have forecasting errors. I believe that by keeping the pre-merger periods in, I was lowering the average quarterly errors, so I was removing smaller numbers.



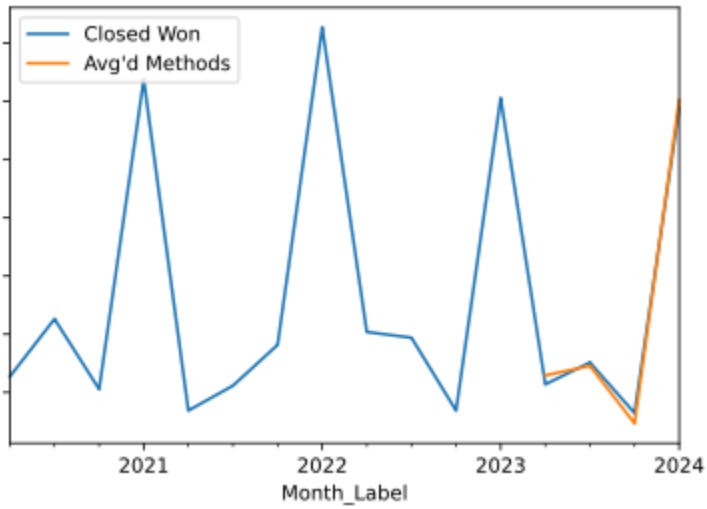
Facebook's Prophet was next, performing better than SARIMA, representing an RMSE of 11.50%. Prophet has the option to make the growth trend linear or logistic. In keeping with Holt-Winters' multiplicative trend, I set the growth parameter to 'logistic'. However, I found that 'logistic' performed worse than 'linear', so I changed the growth parameter to 'logistic' and maintained Holt-Winters' seasonality as 'additive', since that performed better than 'multiplicative'.



Up to this point, I had a lot of prediction lines on my bar graph. I removed some so I could see individual predictions without other distracting forecast lines. I noticed the Holt-Winters forecast over predicted, and SARIMA under predicted. One of the drawbacks of using RMSE as a measure of error is that all numbers are positive, and I couldn't tell which predictions were too aggressive and which were too conservative by that measure alone.



After seeing this, I decided I wasn't too proud to not use an explicit machine learning algorithm to tune my forecast, so I averaged the two predictions, resulting in my closest forecast to the testing data. Over the testing year, the average of SARIMA and Holt-Winters represented an error of 2.96%.



The final RMSEs, again, expressed as a percentage of the total Closed Won for the testing year are as follows:

Method	RMSE %
Average of Holt-Winters & SARIMA	2.96%
Holt-Winters	7.73%
By-Hand Error Removal	8.58%
Facebook Prophet	11.50%
SARIMA	13.15%
Sales Forecast	13.53%

During the presentation of this data to the CFO, she asked if I had this kind of forecast for the current sales pipeline, a prediction of what deals would be Closed Won, and then a sum of those contract values. She requested I stop working on the Time Series analysis portion and start working on the Binary prediction instead, along with a list of similarities between deals that end up in Closed Won and deals that end up in Closed Lost.

Binary Prediction

Unfortunately, I cannot share any images for Binary Prediction since they are tree visualizations with confidential information.

The sales pipeline Binary prediction took significantly more time to prepare and complete than the Time Series. The Time Series prediction just required the quarterly Closed Won numbers and the associated quarterly dates. I sat with my boss and we determined which deal and account level attributes we should include. We found 42 attributes, plus the resulting Closed Won/Closed Lost dependent variable column. Most columns came out of Salesforce as objects, so it took substantial time to set columns to floats, integers, or categories. Several of the 42 columns had missing data that could be imputed on the Salesforce side, allowing us to redownload the dataset with filled values. Other columns, like ‘Billing Country’, had too many blanks for us to reasonably fill in, so we dropped those. The Company recently underwent a transition to a new Salesforce instance, so date fields, like ‘Create Date’, are all set to the date the deals were transferred into the new instance instead of when they were actually created. Fortunately, we found the Salesforce administrators had foreseen people needing that date, and we found a field called ‘Original Created Date’ that had the correct dates. To calculate the age of a deal on its closing date, we subtracted the ‘Original Created Date’ from ‘Close Date’. We also added a conditional field based on the contract value of products. The Company categorizes its clients by the products they purchase, the name of the product if a client purchases a solo product, ‘Suite’ if the client purchases multiple classes of products, or ‘New Product + Suite’ if the client purchases one of the newer products in addition to the multiple classes of products. Other fields we used include how many deals a client has closed with us previously and different marketing fields based on how good a fit we believe they are for us. The original 42 columns turned into several hundred columns when converting the categorical data into dummy variables.

The Company also categorizes its deals by revenue type, ‘New’ deals can be either companies that have no history with us or new business units within an account. ‘Renewal’ deals are returning client contracts for 12 months, and ‘Upsell’ deals are standalone deals where a client wants to add products during an existing contract, generally ending co-terminally with their renewal contract. Renewals may and often do contain embedded upsells, but they are counted as part of the renewal. I found that despite the Binary prediction performing at around 84% accuracy for the entire sales pipeline, it made more sense to split the pipeline into these three ‘New’, ‘Renewal’, and ‘Upsell’ categories.

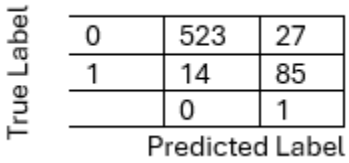
I started the prediction using Elastic Net variable selection for Logistic Regression. I prefer the model simplification ability of LASSO, taking variables down to 0, but I understand that coefficient shrinkage from Ridge Regression is important for correlated terms as well.

I made three regressions, one for ‘New’, ‘Upsell’, and ‘Renewal’. For the ‘New’ regression, 342 variables were shrunk to 0, and 58 were reduced. For the ‘Upsell’ regression, 311 variables were shrunk to 0 and 89 were reduced. For the ‘Renewal’ regression, 302 variables were shrunk to 0 and 98 were reduced. The accuracies from their confusion matrices were 88%, 81%, and 68%, respectively.

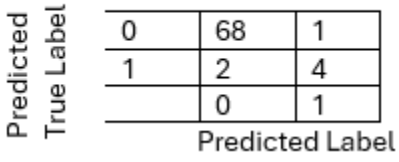
I continued to create a Classification Tree to responsibly see if a better model could be created, and Classification Trees also can include feature importance, an explicit request for this project. The data was already cleaned and prepped, so setting up the trees was easy. Since they are separate, they each have different tunings applied to them. The ‘New’ tree has a cost complexity alpha score of 0.01 and follows the defaults for everything else. The ‘Upsell’ tree has a ccp_alpha of 0.01 as well, but the split criterion performed better as ‘log_loss’ than the default ‘gini’. The ‘Renewal’ tree returns to ‘gini’ but the ccp_alpha is 0.006, much smaller than the other two. These parameters gave the best accuracies when classifying, without overfitting. The Classification Tree went no more than 6 layers deep on any of the three models, and the accuracies are 94% for ‘New’, 84% for ‘Upsell’, and 75% for ‘Renewal’, an improvement on each revenue type.

Revenue Type	Logistic Regression	Classification Tree
New	88%	94%
Upsell	81%	84%
Renewal	68%	75%

When predicting the sales pipelines, I isolated the deal and account names because they can’t help in the prediction, but I also needed them to reattach to the TRUE/FALSE predictions at the end. After reattaching, I added up the contract values of the Closed Won (TRUE) predictions but wanted to compensate for inaccuracies in the predictions. Here is an example of a confusion matrix I had from the Classification Tree ‘New’ prediction:



The accuracy from this confusion matrix is 94%, as mentioned above, but we over-predicted 27 deals, and under-predicted 14 deals. The prediction for this tree with the 75 ‘New’ deals in the pipeline was 70 Closed Lost and 5 Closed Won, but the over and underprediction inaccuracies in the testing data still bothered me. I built a confusion matrix scaler to help build a new confusion matrix based on the pattern of the training and testing data, but for the new 70 and 5. Here is the proposed scaled confusion matrix:



The scaled confusion matrix suggests that roughly following the same pattern as the testing data, one of the predicted Closed Won deals will actually be a Closed Lost, and two of the predicted Closed Lost will actually be Closed Won, so instead of 70 Closed Lost and 5 Closed Won, it will be 69 Closed Lost and 6 Closed Won. I did this for all three revenue types and used the average sales prices in the pipelines to estimate the new Closed Won number, compensated for the testing patterns inaccuracies. Unfortunately, due to confidentiality again, I cannot share the predicted or

compensated pipelines, but the compensated pipeline is more aggressive for ‘New’ and ‘Upsell’, and more conservative for ‘Renewal’, bringing the total compensated pipeline to only 0.23% more aggressive than the Tree predicted by itself.

Results

I recognize that I have gone over the methodological results above, but I am more concerned with the business results of the analysis, the better business decisions made and the more accurate forecast we use for next fiscal year’s planning.

Since there are three different trees, there are three feature importance lists. There is some overlap between the features of each of the revenue types, but they are generally unique from each other. Patterns include:

- more of a certain product on the contract is more likely to be Closed Won
- more history of Closed Won deals and more interaction with us is more likely to be Closed Won, whether it be long standing renewals, or a mix of ‘Upsell’ deals alongside ‘Renewal’ deals
- less history and interaction with us is more likely to be Closed Lost
- certain sales people are more likely to close deals

There was one pattern that raised some concerns in the ‘Upsell’ prediction. We noticed that deals over a certain age get stale, which is expected, and deals under a certain lower age are more likely to be Closed Won, but the age was suspiciously low for the general sales cycle. We believe that the pattern is not a true lead measure, but a lag measure, that salespeople are talking to clients long before they create a contract in Salesforce and only create the contract once they are confident it will close. The salespeople may be talking to their client about an upsell for 60 days, but then create the contract 30 days before it closes because there is more indication of a Closed Won deal, artificially lowering the ‘Age at Closed Won’ field in Salesforce.

I presented this information along with the Time Series predictions to the CFO to a lukewarm reception. My forecasts are much more conservative than what she is used to seeing, even if the machine learning forecast came in far more accurately than the Sales team in the testing data. I had to pivot from, “Here is a more accurate forecast”, to “Here is an early warning system for deals the sales team is working on now and in the future”. She presented the aggregated predictions to the head of Sales along with the CEO and separately, a board member. The board member decided the numbers were too conservative, opting to continue to use the internal forecast from the Sales team. The head of Sales separately agreed that the numbers seemed unrealistic because they disagreed with his gut feeling number, a number that is currently 21.5% higher than my forecast. The CEO recognized the value of the analysis, but also pointed out that with so many newer products and strategies, it is difficult to predict how the changes will be represented by this historical analysis.

Next Steps

The next steps are clear.

1. In the short term, keep working to convince company leadership that accuracy and forecasting to reality instead of to a quota are more effective and efficient for the company than forecasting to a plan number.

2. In the short term, keep working to convince company leadership that integrity in presenting to board members and banks is the morally correct path to take with the knowledge we now possess, and that honesty is better for building relationships than circuitous sales speech around why we keep missing numbers.
3. In the short term, keep adding methods and trying different combinations of methods to see what works the best and reduces the error the most.
4. In the short term, add functionality to access Salesforce through API for automatic data downloads instead of exporting to Excel and reading in those files.
5. In the long term, keep track of my predictions against Closed Won numbers as they come in as proof that basing forecasts on math and historical patterns is more effective than a team trying to look impressive to their bosses.

Conclusion

I believe we are stuck in the Stockdale Paradox, as outlined by James Collins' book *From Good to Great: Why Some Companies Make the Leap...And Others Don't*. Mr. Collins interviews Admiral James Stockdale, a prisoner of war in Vietnam for several years. When asked about how he survived, as well as who didn't, Admiral Stockdale responded with the following:

"The optimists," he replied. "Oh, they were the ones who said, 'We're going to be out by Christmas.' And Christmas would come, and Christmas would go. Then they'd say, 'We're going to be out by Easter.' And Easter would come, and Easter would go. And then Thanksgiving, and then it would be Christmas again. And they died of a broken heart … This is a very important lesson. You must never confuse faith that you will prevail in the end—which you can never afford to lose—with the discipline to confront the most brutal facts of your current reality, whatever they might be."

I do not mean to make light of the impossible situation that Admiral Stockdale and others were placed in. Lives and health are not at stake in this situation. Mr. Collins included the story in his book about business to demonstrate that in a business context, the same attitude about optimism and reality is true. It is a brutal fact of the Company's current reality that the sales forecast is inaccurate at best, and intentionally misleading at worst. However, I understand the CEO can't just announce that there is waning faith in the Sales organization's ability to forecast their pipelines and close deals, because morale would dip, and turnover would jump. Additionally, there is strong faith that the newer products that couldn't be included in the historical trend analysis will offset some of the misses in the aggressive forecast. I do not expect this to be an easy or quick transition in a company where Sales holds so much power. I hope where they beat me in charisma and relationships with company leadership, I can make up for it with hard data and sales history as evidence and impetus for a change in process.

References

Collins, Jim. Good to Great. Random House Business Books, 2001.

Roberts, Ethan Maluhia. “Confusion Matrix Scaler.” ShinyApps, sentimental-post.shinyapps.io/ConfusionMatrixScaler/.

Data Heroes. “Master SARIMAX in Python for Accurate Time Series Forecasting.” YouTube, 9 Aug. 2022, www.youtube.com/watch?v=ySiKZwoTX54&t=308s. Accessed 12 July 2024.

MD1sra Turp. “How to Implement Decision Trees in Python (Train, Test, Evaluate, Explain).” YouTube, 11 June 2021, www.youtube.com/watch?v=wxS5P7yDHRA.

Mulla, Rob. “Forecasting with the FB Prophet Model.” YouTube, 25 Nov. 2022, www.youtube.com/watch?v=j0eioK5edqg. Accessed 28 Aug. 2023.

Paramita. “Holt Winters Model, Easiest Times Series Model. Additive Multiplicative Trend and Seasonality.” YouTube, 17 Dec. 2020, www.youtube.com/watch?v=O6cUkdQeLUQ&t=264s. Accessed 12 July 2024.

“SARIMAX in Python for Time Series Modeling.” Analytics India Magazine, analyticsindiamag.com/topics/sarimax-in-python-for-time-series-modeling/. Accessed 12 July 2024.

“Sklearn.tree.DecisionTreeClassifier — Scikit-Learn 0.22.1 Documentation.” Scikit-Learn.org, 2019, scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

“Sklearn.linear_model.ElasticNet.” Scikit-Learn, scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html.

StatQuest with Josh Starmer. “Classification Trees in Python from Start to Finish.” YouTube, 6 June 2020, www.youtube.com/watch?v=q90UDEgYqeI&t=3480s. Accessed 12 July 2024.