Online Shopping Purchase Intent

ISYE 7406 Project Report

Spring 2022

Group 5

Casey Stanfield, Jia Hui Mo, Mariah Carter, Simon Yee, Xu Anne Zhang

Table of Contents

Introduction

With the advent of the internet, we have seen many traditional industries revolutionized and retail is no different. Online shopping and e-commerce are fast becoming the predominant medium for the exchange of goods and services. It is in the best interest of digital marketers, business owners, and analysts alike to understand factors that contribute to an online purchase being made.

The goal of our project was to predict revenue based on various purchase intent factors measured by activity of a user on an e-commerce website. Our work was designed to predict the binary revenue variable indicating whether there was a sale based on factors such visitor type, information about the region, browser type, whether the visit was during a weekend, if the visit is close to a holiday, the month of the visit, and so forth. The data also deciphered different web page categories as a predictive factor, classifying them as informative, administrative, or product related.

To conduct this analysis, we used a popular data set from the University of California Irvine Machine Learning Repository called the Online Shoppers Purchasing Intention. The data set contains information about a binary dependent variable, Revenue, along with variables such as the number of pages visited per category and other google analytics metrics. We used a variety of modeling methods to predict revenue as well as infer the relationship between revenue and these predictor variables.

As opposed to related literature and research on this dataset and consumer behavior about online shopping in general, our goal was not just to build and tune the data and models with the highest prediction rate - instead our focus was to better understand the relationship between revenue and these predictor variables. Specifically, we wanted to understand the change of y (revenue) over $x_1 \ldots x_n$, and whether ensemble methods were required for a high degree of accuracy in prediction, as we saw in many of the academic articles related to this dataset. We wanted to understand whether a tradeoff of accuracy for interpretability was required, and if so, were there any observable reasons for this.

We modeled the dependent variable with half a dozen different models and explored the data with a handful of other analytical tools. We found that certain variables in the dataset were especially powerful in predicting whether Revenue was generated, in particular the variable "PageValues" which is explained in detail in the data exploration section below. Ultimately, we were able to achieve accuracy rates near 90% with fine tuning of some of the more complex boosting models such as AdaBoost, the details and specifics of which are in the modeling section of this report.

Exploratory Data Analysis

The data we used for this analysis wad provided through University of California, Irvine (UCI) Machine Learning Repository, but per the website, the data was originally taken from C. Okan Sakar and Yomi Kastro, faculty of Bahcesehir University. The data set provides information about purchasing intent of online shoppers. Independent variables in this data provide a spectrum of information, ranging from Google analytics rate data to region, month, special occasions, and time spent on viewing similar products. There are 12,330 rows and 18 columns, with no missing values. A heat map is built (see Figure 1), and there are no notable variables that demonstrate high correlation, with a cut-off of 0.5, with the exception of Administrative_Duration. Variables of potential interest that do not make the cut off are Informational, ProductRelated, ProductRelated_Duration and ExitRates.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	Weekend	Revenue
Administrative																	
Administrative_Duration	0.602																
Informational	0.377	0.303															
Informational_Duration	0.256	0.238	0.619														
ProductRelated	0.431	0.289	0.374	0.280													
ProductRelated_Duration	0.374	0.355	0.388	0.347	0.861												
BounceRates	-0.224	-0.144	-0.116	-0.074	-0.205	-0.185											
ExitRates	-0.316	-0.206	-0.164	-0.105	-0.293	-0.252	0.913										
PageValues	0.099	0.068	0.049	0.031	0.056	0.053	-0.119	-0.174									
SpecialDay	-0.095	-0.073	-0.048	-0.031	-0.024	-0.036	0.073	0.102	-0.064								
Month	0.097	0.058	0.063	0.044	0.156	0.138	-0.067	-0.095	0.067	-0.257							
OperatingSystems	-0.006	-0.007	-0.010	-0.010	0.004	0.003	0.024	0.015	0.019	0.013	0.038						
Browser	-0.025	-0.015	-0.038	-0.019	-0.013	-0.007	-0.016	-0.004	0.046	0.003	0.020	0.223					
Region	-0.005	-0.006	-0.029	-0.027	-0.038	-0.033	-0.006	-0.009	0.011	-0.016	0.024	0.077	0.097				
TrafficType	-0.034	-0.014	-0.034	-0.025	-0.043	-0.036	0.078	0.079	0.013	0.052	0.055	0.189	0.112	0.048			
Weekend	0.026	0.015	0.036	0.024	0.016	0.007	-0.047	-0.063	0.012	-0.017	0.017	0.000	-0.040	-0.001	-0.002		
Revenue	0.139	0.094	0.095	0.070	0.159	0.152	-0.151	-0.207	0.493	-0.082	0.127	-0.015	0.024	-0.012	-0.005	0.029	

Figure 1: Heatmap

Two notable features of this data set are that all numerical attribute related data are quite left skewed. For example, Figure 2 displays the distribution of ProductRelated, a count of how many times a customer viewed a similar product. There are instances of counts that exceed 200, but the majority of them between 0 and approximately 30. The second notable feature is that the dependent column, Revenue, is also very skewed. Revenue contains 10,422 records of instances where a purchase was not made, and only 1,908 instances where a product was purchased (detailed below in Figure 3).

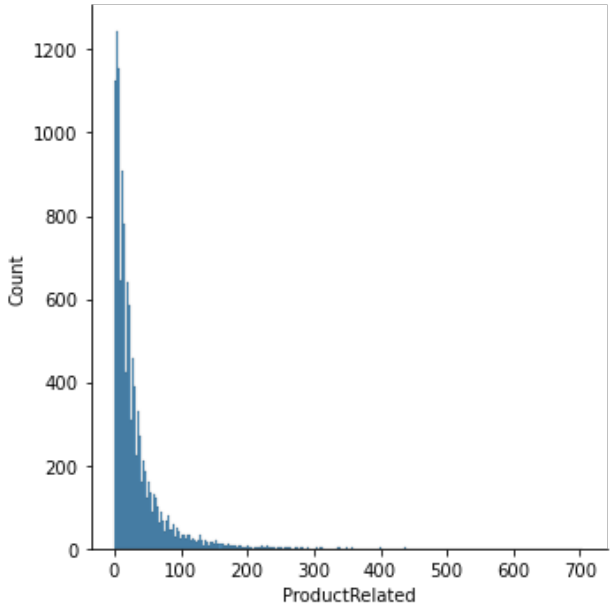

Figure 2: “ProductRelated" Variable Distribution

	Revenue	Count
0	False	10422
1	True	1908

Figure 3: Count of Revenue Values

Supervised and unsupervised principal component analysis transformation were used on the data to understand distributions and patterns between Revenue and predictor variables. Figure 4 displays an unsupervised transformation of the first two principal components. There are no notable patterns, particularly in the direction of the eigenvector, and whether a revenue is made or

not is unclear based on this transformation. On the other hand, using Partial Least Squares (PLS), the data displayed a noticeable difference in direction. Figure 5 displays coefficients from the PLS model: once again, the most impactful coefficient is PageValues. Next are closely tied variables including Month, ExitRates and VisitorType. It should be noted that PageValues is a Google analytics metric that takes revenue into account, and because we have multiple group members running different models, we left it up to the individual whether they wanted to include it, exclude it, or do both.

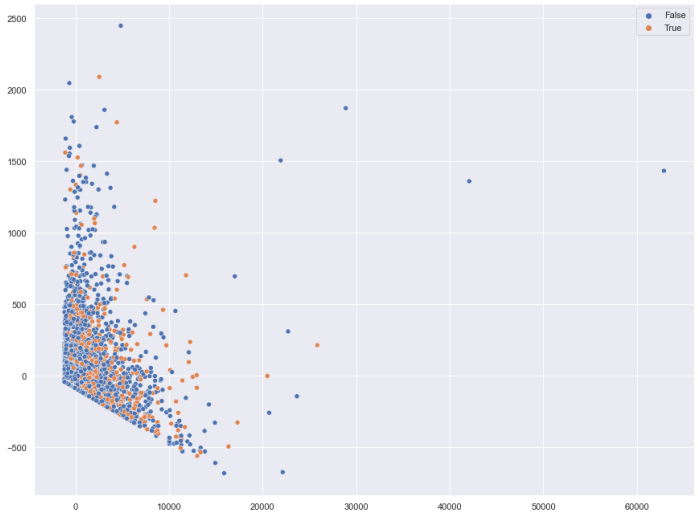

Figure 4: Unsupervised PCA Transformation of Independent Variables

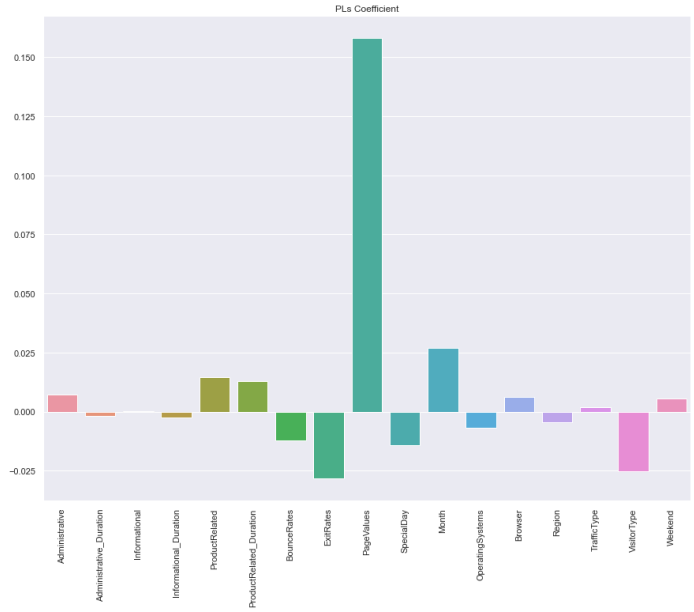

Figure 5: Coefficients of Partial Least Squares

Figure 6 displays a supervised PCA transformation of the independent variables. From this visual, there is a clear group of individuals that do not make any purchases, but it is quite mixed after a certain point. We insert an approximate black line to separate these two groups. An early hypothesis of this transformation leads us to believe that, given how influential PageValues is, generally after a certain point no matter how many page visits a customer make, a distinguishing point in whether a purchase is made or not is decided by factors like time of the month. This is true whether they're a new or returning customer, and other factors like how many related items they viewed, and how much time is spent viewing these related products. This complexity may be an early visual explanation of why ensemble methods such as Random Forest (RF) and Boosting are heavily used in academic literature for this data.

Figure 6: Supervised PCA Transformation of Independent Variables

As discussed in the previous section, since PageValues' formula contains revenue, another PLS transformation is ran without it. In Figure 7, there continues to be overlap with those who made purchases and those who did not, but this model suggests that ExitRates, VisitorType, and month to be the largest coefficients, closely followed by ProductRelated, ProductedRelated_Duration, BounceRates (percentage of visitors who enter and then immediately leave the site) and whether it was a special day or not.

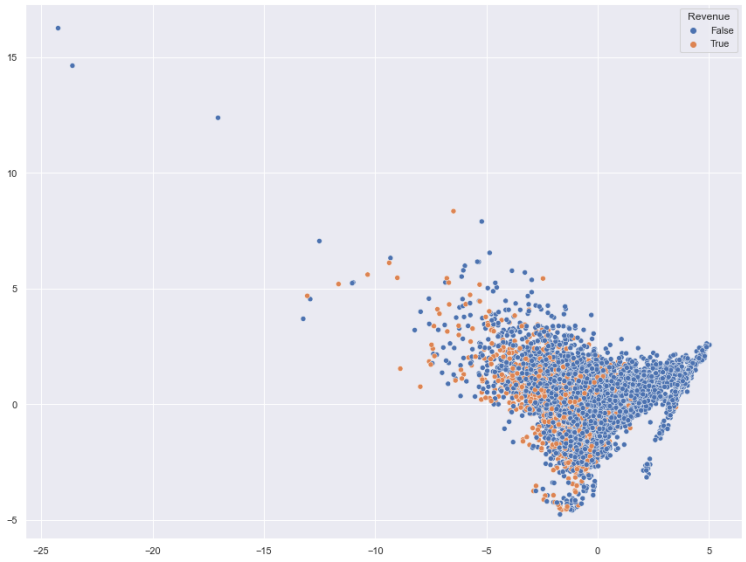

Figure 7: Supervised PCA Transformation of Independent Variables (No PageValues)

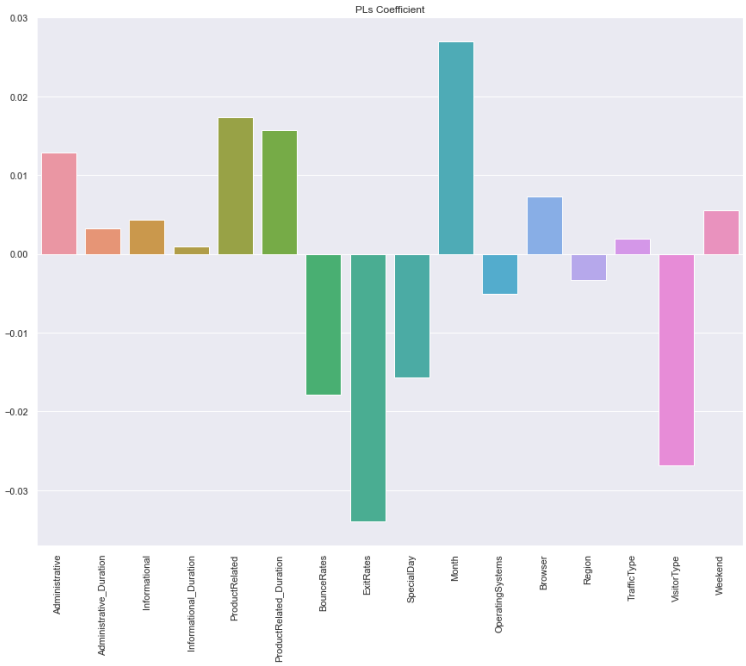

Figure 8: Coefficients of Partial Least Squares (No PageValues)

Methodology

Logistic Regression

Logistic regression is a supervised classification algorithm that can be used to predict the probability of event success and event failure. It is used when the target variable is binary, or it can extend to multiple classes, making it a perfect fit for our Revenue dependent variable. Rather than using more mathematically complex models, logistic regression is often the top pick for benchmark models to measure model performance because it is the simplest machine learning algorithm and easy to implement. The predicted parameters give inference about the importance of each feature, so it can easily use the model to find out the relationship between each predictor and the dependent variable.

Additionally, logistic regression assumes that there is minimal or no multicollinearity among the independent variables. To handle multicollinearity, we used stepwise regression by AIC in both directions to see if there is an improvement in model accuracy. By comparing the full model and reduced model, the result of the test for subsets of coefficients suggested that the coefficient of Administrative_Duration, Informational_Duration, and Region are equal to zero. From the result of 10-fold cross-validation, we found that model accuracy was much improved. Additionally, we found that PageValues, ExitRates, and ProductRelatedDuration were the most useful variables for prediction.

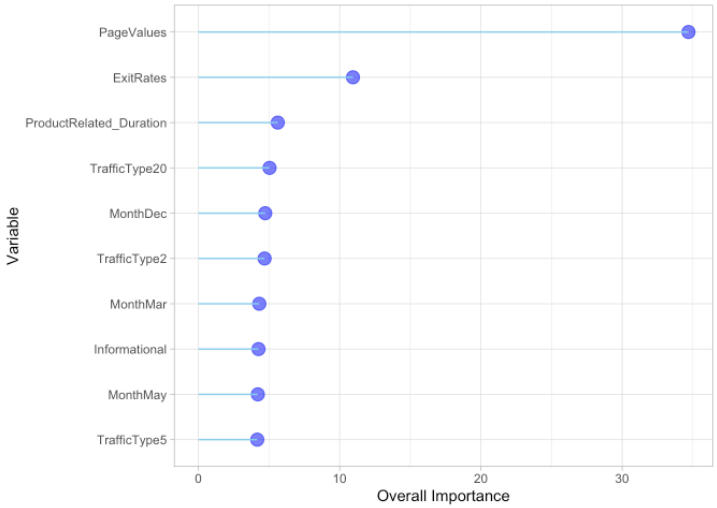

Figure 9: Importance of Each Variable

Since the logistic regression model returns a probability between 0 and 1, we had to pick a threshold for a decision boundary to predict which class each test data point belonged to. The setting of the threshold is very important for interpreting the results of a logistic regression model correctly (and is dependent on the classification itself). To maximize correct classification on our training and testing datasets, we wanted to find a model that was able to perfectly classify those data points with high accuracy. Ideally, both precision and recall are very close to 1, although we acknowledge this is unlikely.

We found our logistic regression model to reach the highest accuracy when the threshold was set at 0.75, whereas it has more false positives compared to the default value of 0.50. For use in a business context, we would recommend a higher threshold to avoid false positive cases that could lead to erroneous interpretations of results - for instance, the model might predict that the company would earn revenue, but if customer behavior does not match the results of such a model, the company might end up with overloaded inventory, which could have negative effects on the business. This could take the form of inventory costs or other hidden costs, all of which could result in negative impacts on financial statements. For these reasons, given the context of the data and who would be consuming such a model in a real business context, we want to choose a decision value to reduce the number of false positives for our model. The chart below details the relationship between a threshold and the prediction accuracy: we found the accuracy of our model to be maximized at a threshold value of 0.75, where the accuracy of the model was approximately 88%.

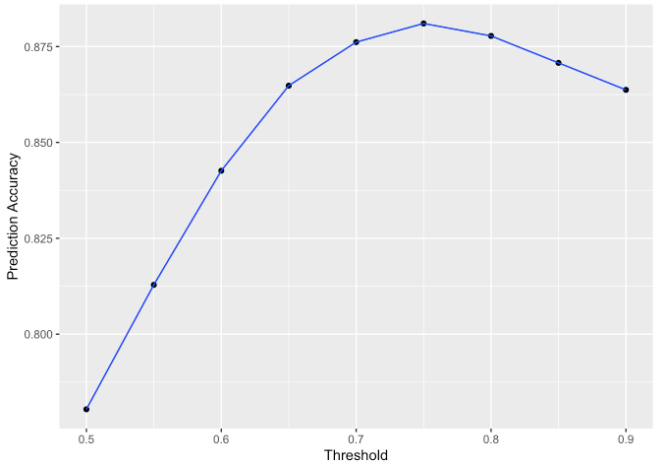

Figure 10: Relationship Between Threshold and Prediction Accuracy

Naive Bayes

Naive Bayes is a family of probabilistic classifiers based on Bayes Theorem. Assumptions for these models are that our predictor variables are independent and equally contribute to the dependent variable. Our dataset does not meet these assumptions but there has been some research that has shown that Naive Bayes models can perform well despite this. For our analysis, we focused on three Naive Bayes algorithms: Gaussian, Bernoulli, and Multinomial.

Gaussian Naive Bayes is a classifier that is used when predictors are continuous, and a Gaussian distribution is assumed for the predictors. *Bernoulli Naive Bayes* algorithm is preferably used when the predictors are boolean in nature and are assumed to follow a Bernoulli distribution. The final type of Naive Bayes model tested is *Multinomial* which assumes that there are differing distributions for each predictor in the dataset. Neither Gaussian nor Bernoulli model assumptions hold with our dataset but an observation of the Naive Bayes model is that they can often perform better than expected despite these unmet expectations.

After running these three models on our dataset, we found that none of the Naive Bayes models performed well with default parameters. In particular, the Bernoulli model blindly assigned all data points to the majority class, which was no revenue. Tuning was done primarily with manual feature selection to remove interdependence from the predictors based on their correlation found during our EDA and by tuning the smoothing variable for both “Gaussian” and “alpha” in the Bernoulli and Multinomial models. The feature selection portion involved using the full dataset then removing the "duration" predictors while leaving the page count predictors. For example, we removed the ProductRelated_Duration variable but kept the ProductRelated variable, as these two have a very strong relationship. After this failed to achieve the performance of Logistic Regression, a second subset was created by removing the dummy variables created for the original VisitorTypes predictor. We found that all variables related to the PageValues variable in an interdependent way, as all page related data and the ExitRates predictor had the highest correlation with the target variable Revenue.

LASSO

LASSO is a regression analysis that aims to fit the data by minimizing the sum of residual squares with lambda penalty terms. The goal is to simplify the regression equation through a penalty term (lambda) that aims to keep relevant variables through shrinking coefficients and by keeping variables that maintain the largest sum of squared residuals. The larger the penalty term, the heavier the penalty (coefficients becoming close to 0). The data is first standardized, and then a Monte Carlo cross validation method is applied, where we split the training and testing data in 80/20 split randomly in each iteration. We ran this model a total of 1,000 times.

It should be noted that our problem is a classification problem, and technically there formally is not an obvious application of the LASSO logistic regression since LASSO uses the least squares method. Our alternative is a logistic regression with L1 penalty terms, which should be nearly equivalent. Two modes are created, one with all predictors and another with all predictors except PageValues; the models averaged roughly 88.2% and 84.4% respectively. Models with all predictors show PageValues to be the largest coefficient, indicating a larger likelihood revenue is made as PageValues gets larger, which is consistent with the formula. While the model without PageValues shows ExitRates to be the largest coefficient (see Figure 12).

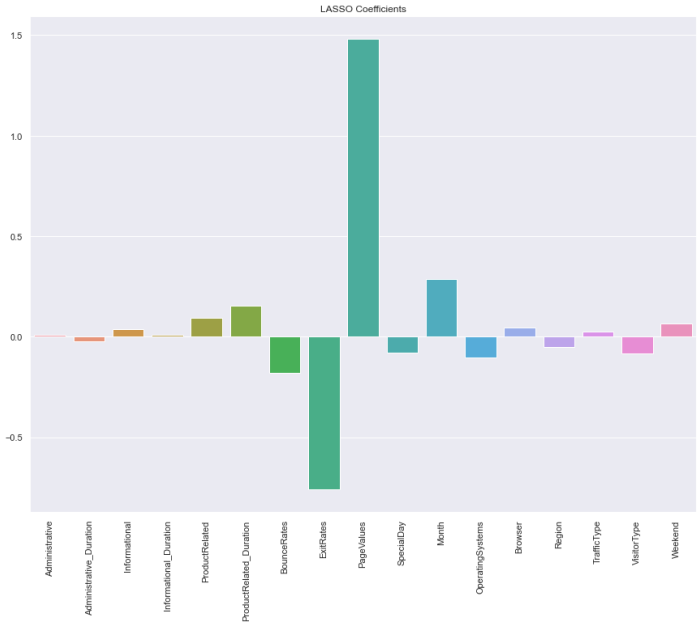

Figure 11: LASSO Coefficients of Variables

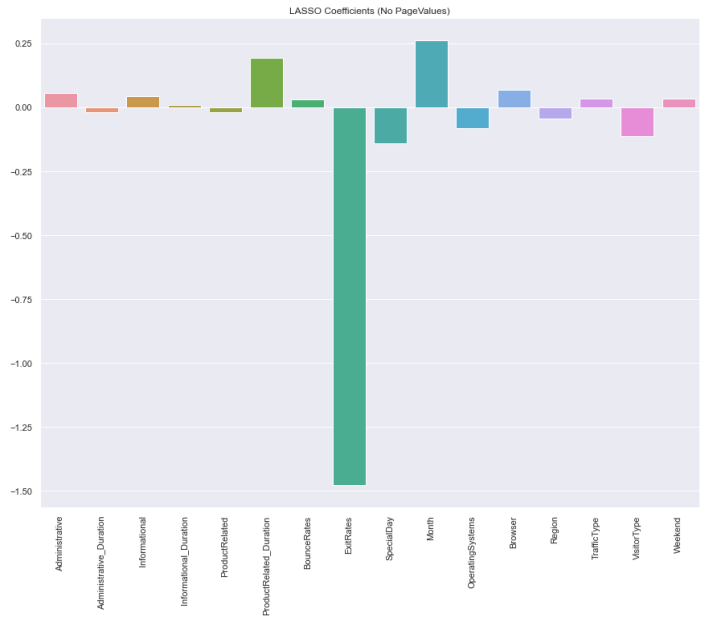

Figure 12: LASSO Coefficients (No PageValues)

Discriminant Analysis

Discriminant analysis is another common supervised learning methodology used for classification. In discriminant analysis, the target variable is separated into different classes. By minimizing the variance within classes and maximizing the variance between classes, the model finds the linear combination of the dependent variables. Linear discriminant analysis (LDA) is used when a linear boundary is required between classifiers, and it follows the Bayes rule under the assumption of normality for the distribution of observations in each response class with class-specific mean and common variance.

LDA can also be used as a dimension reduction tool. LDA and PCA are very similar in the sense of dimension reduction. On the other hand, LDA accounts for the differences between classes since it is a supervised method. Unlike LDA, quadratic discriminant analysis (QDA) is used to find a non-linear boundary between classifiers. Like the assumption of LDA, the distribution of observation in each class has class-specific mean and covariance. Since LDA has fewer parameters to estimate, it is less flexible than quadratic discriminant analysis. When the training observations

are larger and the target classes have very different covariance matrices, LDA suffers from high bias and QDA would be a better approach.

Our application of LDA to the Revenue dataset used a Monte Carlo cross validation with an 80/20 training and testing split 100 times and compared the result for LDA and QDA. We found the following variables to have significant differences in group mean: Administrative Duration, Informational Duration, Product Related, ProductRelated_Duration, and PageValues. Unsurprisingly, when we fitted a model with all these variables, model accuracy improved significantly. We also found that compared with QDA, a linear classifier is more suitable for our dataset since it has higher model accuracy. The output of our LDA model clearly showed the separation between these two groups (the binary response variable Revenue) were quite close with lots of overlap.

AdaBoost

AdaptiveBoosting (AdaBoost) is a common boosting algorithm that can perform quite well on these kinds of problems but has some drawbacks – thus, we chose to include it as a baseline ensemble method. AdaBoost works by repeatedly training models on a dataset then, through weighting and penalization, aggregating the results into a final model which should be superior to the individual models. A benefit of this approach is that a better performing model can often be achieved with less overfitting which should yield a model that performs better on new datasets. Unfortunately, AdaBoost is often slow to train as there is a tendency to attempt to fit extreme observation values.

Through our implementation of AdaBoost, we found that it achieved nearly the same accuracy as XGBoost (detailed findings of below) at about 87% but it took 3-4x as long to train, and nearly 5x as long to train as the Naive Bayes model. We believe the AdaBoost model did not deliver superior performance due to the known issue with its performance being hindered by exploring extreme predictors in observations and using those as important features of the dataset: more specifically, the untuned model on the full dataset had an accuracy of 16% from attempting to assign all observations to the minority (where Revenue == TRUE). Additionally, the accuracy

of the model was only changed by 1% when using oversampling to address the class imbalance of the dependent variable Revenue.

XGBoost

Extreme Gradient Boosting (XGBoost) is an open-source library that implements a regularizing gradient boosting framework. It has been very popular with a variety of real world and competition implementations with quite positive results. The library provides several built-in tuning features to improve upon performance of other boosting implementations as well as a hefty speed increase. The model uses a Newton-Raphson method for gradient boosting. This basically functions by making a random guess at the x-value then making a tangent line at f(x) for this x. Then the intercept of the tangent line is used as the starting point of the next iteration of the process until finally f(x)=0 is found. XGBoost also implements automatic feature selection, an extra randomization parameter, a proportional shrinking of leaf nodes, and a unique penalization of trees.

XGBoost, with default parameters, performed roughly equally to our tuned AdaBoost model. To improve upon this, the number of estimators, learning rate, depth, cross tree sampling, and evaluation metric were iterated upon to find a best performing collection. With the speed of this model training, which was only on average twice as long as the Naive Bayes model, these iterations were done quickly.

Conclusion & Findings

Over the course of the past several weeks, we have tested half a dozen models on the shopping dataset: ultimately, we found that the XGBoost model performed the best, resulting in close to a 90% accuracy rate after tuning the model parameters. A few other noteworthy findings to highlight from some of the other models we tested are below. Notable top 3 variables of each model are listed in the table below:

Model	X1	X2	X3
Logistic Regression	PageValues	ExitRates	ProductRelatedDuration
Naive Bayes (unordered)	PageValues	ExitRates	
LASSO	PageValues	ExitRates	Month
LASSO (alternative)	ExitRates	Month	ProductRelatedDuration
LDA (unordered)	PageValues	Administrative Duration/Informational Duration	ProductRelated/ProductRelatedDuration
AdaBoost	PageValues	Month	
XGBoosting	PageValues	BounceRates	Month

Figure 13: Summary Table of Top Variables by Model

Through our exploratory and modeling phase, we were unable to determine an exact explanation and evidence for cutoffs or ranges, but we were able to develop potential hypotheses that are supported by multiple models. As an example, if we exclude PageValues due to its conflict of having revenue in its formula, we can understand that in general, if a customer quickly leaves your website without browsing, the greater the possibility they will not make a purchase, implying and explaining the importance of the variable ExitRates. This means there is a lot of incentive to keep the customer on a website as along as possible, and as such, we would recommend a better recommendation system to help customers see similar items, and/or better interactions and design that helps keep customers engaged. Some of these hypotheses are further supported by appearances of ProductRelated and ProductRelated_Duration, which suggests that keeping your customers browsing for a longer period of time may increase their possibility of a purchase. On the other hand, it is hard to determine if appearances of ProductRelated or ProductRelated_Duration come from customers shopping around for best prices.

Appearances of other variables like Month may indicate shopping patterns, discounts, or holidays of customers, although the latter and others would be contradicted by the lack of appearance of collected variables like weekend (Yes/No) and special holidays (Yes/No). We

recommend that date is further investigated by future researchers, and as for companies and other practitioners, we recommend another date level analysis like day/week level as opposed to month to prevent this level of uncertainty. AdministrativeDuration is currently an unquantifiable variable for our team, we simply believe those who will spend time updating addresses, profiles, and modifying their credit cards are taking the time to do so because of their status as regulars. There is a lack of overall evidence for this, because returning visitors suggested by relevant models are least likely to return. In conclusion, we understand the popularity of ensemble methods due to high accuracy and flexibility, because it is significantly easier finding and tuning a model that works, rather than discovering why it works. Despite this, we recommend inference at a minimum level and not to do so just because it works; while the accuracy may be high, nothing may be measured.

Lessons Learned

We discovered that depending on your background, bias, and knowledge of the topic, the application and interpretation of an individual methodology in variable selection and the resulting findings may vary greatly. It is challenging to reach a consensus on the root cause of an observable outcome, especially when the finding is subjective, like why a customer may view similar products or many products. Are they shopping around for feedback on reviews? Is it price comparisons? Are they waiting for a discount? Are they checking if it is the right tool for the job? Or perhaps the correct replacement part? We have no right answers to this question, and depending on the previously discussed items, an individual's answer and recommendation will vary greatly as well. While the process certainly requires many approaches and iterations, we found that there are certainly explanations that make more sense than others, and that it is important that we take both a strong quantitative and qualitative approach to current and future data science projects.