STSCI 4740 Final Project

Analysis of the Revenue of Online Businesses

Chloe Qiu (jq99)

Lu Cao (lc892)

Josie Zhuo (qz328)

Jiadi Huang (jh2649)

Jiali Liu (jl2685)

May 2021

Introduction

In the past few years, online shopping, which brings convenience, mostly free shipping and lower price, has become an indispensable part of the global retail framework. With the advent of the Internet and the continuous digitization of modern life, the number of online shoppers is increasing every year. Our group will dig further into online shopping from a statistical perspective. In this project, we will focus on the dataset "online_shoppers_intention.csv". To predict the revenue of online businesses based on the online shoppers' viewing and purchasing behaviors, features of pages in the e-commerce sites, and indicators of other factors, we will use multiple machine learning models including Random forest, Logistic regression, and KNN.

Dataset Description

We attempt to answer our research question by analyzing data from Online Shoppers Purchasing Intention Dataset provided by UCI Center for Machine Learning and Intelligent Systems. The dataset "online_shoppers_intention.csv" consists of 12,330 observations with 18 variables. We set Revenue as our response variable and the other 17 variables as our predictors. From the correlation matrix in Figure 1, we can see that there are high correlation between "BounceRates" and "ExitRates", "Informational" and "Informational_Duration", "Administrative" and "Administrative_Duration", and "ProductRelated" and "ProductRelated_Duration". Since they are measured based on different criterias in all cases, we will keep all of these variables but at the same time keep in mind the high correlation. In Table 1, the description and type of each variable and how it is cleaned are specified. Plus, we added a predictor "KeepScore" derived from "ExitRate" to rate how the page can keep the visitors to trigger further page views.

***Figure 1**: Correlation matrix of the variables in dataset*

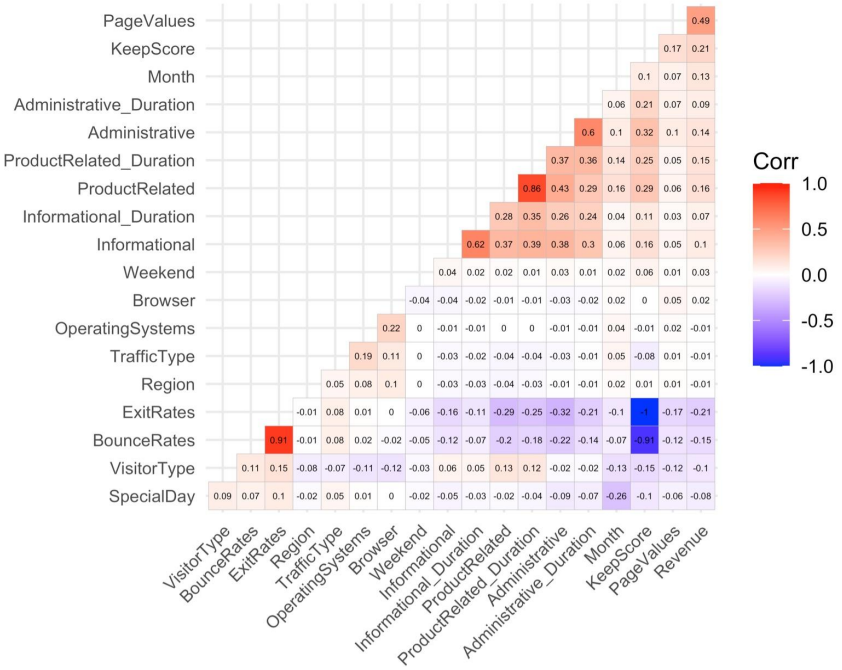

***Table 1**: Description of variables obtained from dataset*

Variable	Description	Additional Information
Administrative	the number of administrative pages visited by the visitor in that session	Mean: 2.32 Standard deviation: 3.32
Administrative_Duration	total time spent in administrative pages	Mean: 80.82 Standard deviation: 176.781
Informational	the number of informational pages visited by the visitor in that session	Mean: 0.50 Standard deviation: 1.27
Informational_Duration	total time spent in informational pages	Mean: 34.47 Standard deviation: 140.75
ProductRelated	the number of product related pages visited by the visitor in that session	Mean: 31.73 Standard deviation: 44.48
ProductRelated_Duration	total time spent in product related pages	Mean: 1194.80 Standard deviation: 1913.67
BounceRates	the percentage of visitors who enter the site from that page and then leave without triggering any other requests to the analytics server during that session	Mean: 0.022 Standard deviation: 0.048
ExitRates	the percentage that were the last in the session for all pageviews	Mean: 0.043 Standard deviation: 0.049
PageValues	the average value for a web page that a user visited before completing an e-commerce transaction	Mean: 5.89 Standard deviation: 18.57
SpecialDay	the closeness of the site visiting time to a specific special day in which the sessions are more likely to be finalized with transaction	The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date.
Month	Month of the year	Integers from 1 to 12 represents Month from January to December
OperatingSystems	Operating systems Type	Categorical
Browser	Browser Type	Categorical
Region	Region Type	Categorical
TrafficType	Traffic Type	Categorical
VisitorType	Visitor Type	Convert "Other" to integer 1, "New_Visitor" to integer 2, "Returning_Visitor" to integer 3.

Weekend	Indicating whether the session takes place on weekend	Boolean
Revenue	Indicating whether the session produces revenue or not	Boolean
KeepScore	Keep Rate = 1 - Exit Rate; Keep Score = Normalized Keep Rate (scaled to 0 to 1)	A score from 0 - 1

Modeling Methodologies

I. Random Forest

The random forest model can solve classification and regression problems. The algorithm creates a forest with a number of decision trees. The trees in the forest, the more robust the model. Multiple trees are being generated in this algorithm to classify a new object. Each tree then votes for a classification and then the forest chooses for the classification that gets the most votes from the trees. Random forest can handle high dimensionality because of its characteristics by growing multiple trees. Random forest has a tendency to overfit classification problems when the dataset is very noisy. However, the data is mostly cleaned and shouldn't have a problem with overfitting problems.

After fitting the dataset, the random forest model produces a good overall prediction of the revenue variable. The out of box estimate of error rate is only 10.41%. However, the model does not do a good job of estimating value 1 which is when the customer brings revenue to the site. While predicting 0s, the model is pretty accurate. It has an accuracy rate of 95.8%. However when it comes to predicting 1s, the model only has an accuracy rate of about 60%. By plotting the model, we can also see that the model stabilizes around 50 trees and stays constant after that.

Figure 2*: The mean decrease accuracy table and the mean decrease gini table*

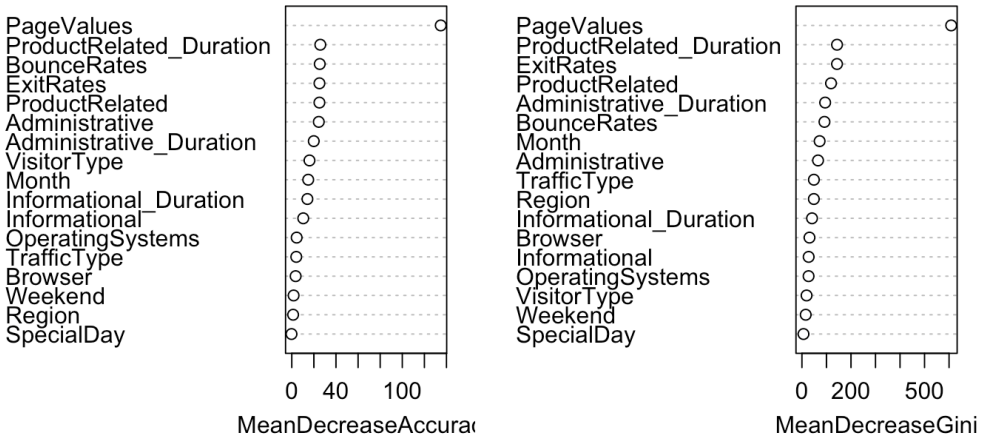

Figure 2 shows that the variable PageValues is very important to the model, and removing the pageValues variable would result in an additional misclassification of about 130 observations on average. All the other variables are not as important looking at the Mean Decrease Accuracy plot. From the Mean Decrease Gini Graph, the PageValues variable is also the most important one. The variable contributes to the homogeneity of the nodes in the model by the most. Although the ranking of other variables have changed, it is not significant since all the gini values for the rest of the variables are very close to each other, just like the accuracy values.

The test set produces similar results to the out of box estimate, the model produces high test accuracy with a low error. However, the false positive rate is pretty high, with a value of 0.4176. The false positive rates and false negative rates are also similar.

***Figure 3**: The Misclassification Rate*

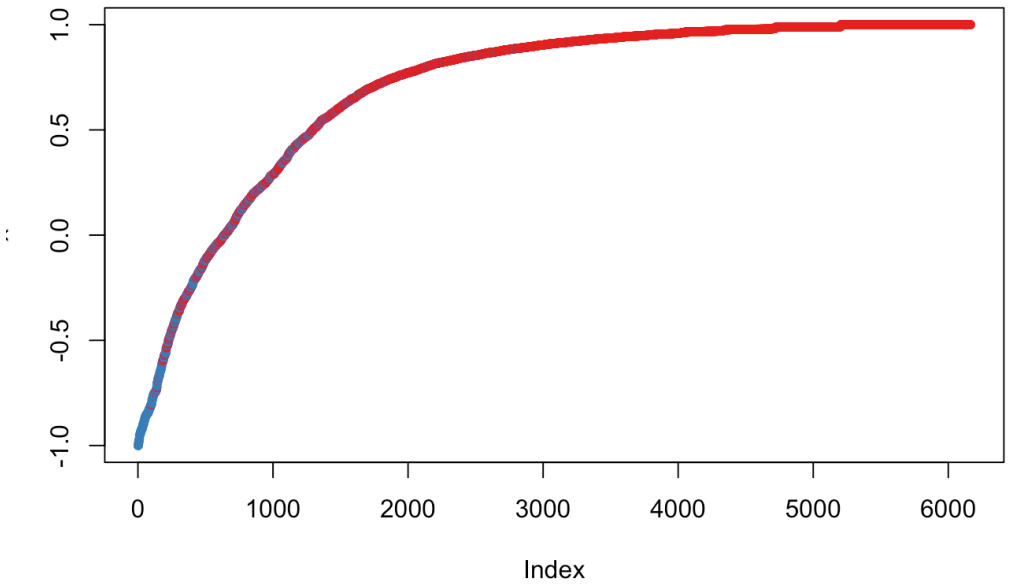

Figure 3 shows the misclassification rate. The blue dots represent the values 1 and red ones represent the values 0. Negative values represent inaccurate predictions and positive values represent the accurate predictions. We can see from the graph that the positive ones are mainly 0s meaning the 0 predictions are accurate while the 1s are not.

In conclusion, the random forest model has a high accuracy score for the overall prediction. It does a good job of predicting the customers that would bring no revenue to the site with a testing false negative rate of 0.038. However, the model produces a high false positive rate of 0.4176. The model does not overfit, which is a concern for random forest models when it comes to classification models. The out of box error rate matches with the test error rate. From the model, we can also see that the most important variable is PageValues from both the accuracy table and the mean decrease in the gini table. It has much more impact than all the other features.

II. Logistic Regression

Logistic regression is another method to find the relationship between the qualitative response and its predictors. It belongs to the class of generalized linear modeling and is used to predict categorical (binary) target variables. This is achieved through a logistic function which takes values between 0 and 1, and always produces an S-shaped curve.

As for this question, we firstly build a full model,

glm.fit=glm(Revenue~.,family="binomial",data=Online),

which contains every predictor variable in R using the glm() function. After fitting the model with the dataset and running the summary, it shows that some variables are insignificant in predicting the Revenue. By setting the critical value to 0.05, it turns out that, of all the predictors, ProductRelated, ProductRelatedDuration, ExitRates, PageValuesSpecialDay, Month, VisitorType and Weekend are statistically significant as their p-values are less than 0.05. Among them, ExitRates, PageValues and Month are corresponding to the lowest p-values, indicating that they have relatively strong relationships with the response.

In order to make the model simpler and more interpretable, we remove those insignificant variables by setting the corresponding coefficient estimate to zero and build a new reduced model,

glm.fit1=glm.fit=glm(Revenue~ProductRelated+ProductRelated_Duration+ExitRates+PageValues+SpecialDay+Month+VisitorType +Weekend,family="binomial",data=Online),

which only contains the significant predictors yielded. From the summary of this new model, we can conclude that each of the variables are predictive thus useful.

To assess the performance of this regression model, we compute the test error. To calculate the test error of the model, we divide our data set into training data and test data to avoid over-fitting, which might happen if the same data set is used for both training and testing. Here we equally split the data into two categories with n/2 observations in each one. We then form the logistic regression function using only training data and assess its accuracy based on test data.

By computing the predictions of test data and comparing them to the actual observations, we may conclude via the confusion matrix that the percentage of correct predictions on the test data is (5110+343)/6165, equal to 88.45%. In other words, the test error is 11.55%. Apart from this overall accuracy, we can also get the followings: True Positive Rate (TPR) = 343/(343+131) = 72.3%, indicating 72.3% positive values, out of all the positive values, have been correctly predicted. False Positive Rate (FPR) = 131/(131+5110) = 2.5%, indicating 2.5% negative values, out of all the negative values, have been incorrectly predicted. True Negative Rate (TNR) = 5110/(5110 +581) = 89.8% , indicating 89.8% negative values, out of all the negative values,

have been correctly predicted. False Negative Rate (FNR) = 343/(343+581) = 37.1%, indicating positive values, out of all the positive values, have been incorrectly predicted.

```
glm.pred    0    1
       0 5110  581
       1  131  343
[1] 0.8845093
```

III. Other Classification Analysis

1. Linear Discriminant Analysis (LDA)

We first use the LDA method as one of the classification analyses. LDA is used to find a linear combination of features that characterizes or separates two or more classes of our final object, the revenue. Generally speaking, it is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order to avoid overfitting and also reduce computational costs. After fitting the dataset, the LDA model produces an overall prediction of the revenue variable with a test error of 0.1186.

Specifically, it gives True Positive Rate (TPR) = TP/(TP+FN) = 0.8892, which indicates 88.92% positive values, out of all the positive values, have been correctly predicted; False Positive Rate (FPR) = FP/(FP+TN) = 0.2351, which indicates 23.51% negative values, out of all the negative values, have been incorrectly predicted; True Negative Rate (TNR) = TN/(TN+FP) = 0.7686, which indicates 76.86% negative values, out of all the negative values, have been incorrectly predicted; and False Negative Rate (FNR) = FN/(FN+TP) = 0.1108, which indicates 11.08% positive values, out of all the positive values, have been incorrectly predicted.

2. Quadratic Discriminant Analysis (QDA)

The second method we apply is QDA. It is a variant of LDA in which an individual covariance matrix is estimated for every class of observations. It provides an alternative approach to assume that the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all K classes. Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction. However, unlike LDA, QDA assumes that each class has its own covariance matrix. QDA is particularly useful if there is prior knowledge that individual classes exhibit distinct covariances. After fitting the dataset, the QDA model produces an overall prediction of the revenue variable with a test error of 0.1667.

Besides that, it has a True Positive Rate (TPR) = TP/(TP+FN) = 0.9306, which indicates 93.06% positive values, out of all the positive values, have been correctly predicted; False Positive Rate (FPR) = FP/(FP+TN) = 0.5198, which indicates 51.98% negative values, out of all the negative

values, have been incorrectly predicted; True Negative Rate (TNR) = TN/(TN+FP) = 0.4802, which indicates 48.02% negative values, out of all the negative values, have been incorrectly predicted; and False Negative Rate (FNR) = FN/(FN+TP) = 0.0693, which indicates 6.93% positive values, out of all the positive values, have been incorrectly predicted.

3. K-Nearest Neighbors (KNN)

Finally, we choose KNN to further help predict revenue in the dataset. KNN relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data. It aims to estimate the conditional distribution of Y given X, and then classify a given observation to the class with the highest estimated probability. Despite the fact that it is a very simple approach, KNN can often produce classifiers that are surprisingly close to the optimal Bayes classifier. The choice of K has a drastic effect on the KNN classifier obtained. When K is small, the decision boundary is overly flexible and finds patterns in the data that don't correspond to the Bayes decision boundary. This corresponds to a classifier that has low bias but very high variance. As K grows, the method becomes less flexible and produces a decision boundary that is close to linear. This corresponds to a low-variance but high-bias classifier.

After fitting the dataset, we first select the optimal K to give the best performance of the model, which is 13, with a test error of 0.1345. Moreover, it has True Positive Rate (TPR) = TP/(TP+FN) = 0.9860, which indicates 98.60% positive values, out of all the positive values, have been correctly predicted; False Positive Rate (FPR) = FP/(FP+TN) = 0.2885, indicating 28.85% negative values, out of all the negative values, have been incorrectly predicted; True Negative Rate (TNR) = TN/(TN+FP) = 0.7115, indicating 71.15% negative values, out of all the negative values, have been correctly predicted; False Negative Rate (FNR) = FN/(FN+TP) = 0.1279, indicating 12.79% positive values, out of all the positive values, have been incorrectly predicted.

By comparing the indexes among the three methods, we can conclude that LDA gives the best prediction with minimum test error, and it does a good job of predicting the customers that would bring no revenue to the site with a testing false negative rate of 0.2351.

Discussion

I. Comparison

We compare the test error rate for each of the methods discussed above.

***Table 2**: Test error for the each predicting methods*

	Random Forest	Logistic Regression	LDA	QDA	KNN
Test Error	0.1041	0.1155	0.1186	0.1667	0.1345

As shown in Table 2, among all the methods, the test error rates of all the methods are between the range of 0.1 to 0.2. the Random Forest method has the lowest test error rate, but is unstable when predicting the 0's and the 1's. QDA and KNN have higher test error rates relative to the other methods, but the difference is not significant and could be due to the specific training and test data.

II. Conclusion and future work

In this paper, we have studied the performance of different learning algorithms on the data of online shoppers. The goal of our work was to identify a suitable model that can predict the purchase intention of a shopper more accurately. In general, we derived four models that include all the predictors in the dataset, and one model that uses less predictors while achieving higher accuracy by comparing the significance levels. We are also able to filter the driving factors of "Revenue" through key features such as feature importance in random forest. Plus, we employed multiple criteria and visually inspection of the data to assess the performance of each model. We finally draw the conclusion that the Random Forest method is relatively the most accurate model with the lowest test error rate.

However, the comparison among test errors of each model could only provide us a general idea of the optimal model to choose. Therefore, more benchmarks could be used in the model performance evaluation. We could have done cross validation for each model and compared the results. Though we have got a decent accuracy for this dataset, the accuracy can be improved by using other deep learning methods. Besides, more data sets collected in other time periods can also be included, which could lead to an increase in the accuracy as we have experimented with a very diversified and sparse dataset.

In order to improve the random forest model, we can try bagging and boosting to see if those techniques provide us with a better model. By using bagging, we can get a less complex decision boundary, which could prevent overfitting from happening. By using boosting, we increase the model complexity and aim to decrease the bias of the model. As for the two discriminant analysis models, the predictors are very likely to be correlated with each other, which would affect the performance of the predicting models. Therefore, methods like predictor normalization can be applied to the data in order to increase the performance of LDA and QDA models.

Reference

Data: Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018).