Online Shopper Decision

Authors:

Yilin Sun, Victoria He

Submitted to:

BUSN 41201

June 6, 2022

We pledge our honor that we have not violated the Honor Code during the preparation of this assignment.

Abstract

This project analyzed 12,330 sessions of online shopping data featuring actions of shoppers and information about the webpages, in an attempt to explore modeling options to yield predictions about whether an online purchase is made or not, and to gain some insight to what factors have most significant influences on the revenue of online shopping websites. During the analysis process, we implemented dimension reduction with LASSO regression and FDR analysis, before attempting logistic regression and tree-based models, which are the natural options given our dependent feature from this model is categorical and binary. After realizing that tree-based models are not ideal for prediction due to the overwhelming effect of over-fitting as a result of the large amount of samples, we expanded our model selection options by turning to Naive Bayes models. After fitting general Naive Bayes, Gaussian Naive Bayes, and adding kernels to the models, we discovered that the general Naive Bayes model has best performance in terms of prediction, reaching a 99% out-of-sample prediction accuracy. In addition, our logistic regression model and tree-based models suggested that the page rank values of the shopping website, the month when the shopping session happened, and the Bounce Rate of the shopping website contributed the most to a shopper's decision of making a purchase. (6,)

Contents

Chapter 1

Introduction

1.1 Background

Online shopping has become an essential part of our daily lives, especially since the outbreak of the pandemic. The rise of online shopping is so rapid that the global online shopping market have reach nearly 4 trillion dollars in 2020. In the United States alone, it is predicted that there will be 300 million online shoppers in 2023, which is 91% of the current population in the U.S.(3,) However, it is not necessary that all shoppers browsing the online shopping websites would finally make a purchase: it is highly possible that they are only "window shopping", or simply having a look online and buying the items in other places, such as a local store. Therefore, what drives online shoppers to decide on a purchase from a particular website become vital to online shopping merchants, and has been a subject of study in many different fields. In the following sections, we will be exploring what factors affect shoppers' decisions and how to predict their decisions.

1.2 Introduction to Dataset

Our dataset, **Online Shoppers Purchasing Intention Dataset**, is an open-source dataset from University of California Irvine's Machine Learning Repository. This dataset records 17 features of 12330 shoppers, in addition to if an online purchase was made or not ("Revenue" feature). These features are defined as follows: (6,)

- **"Administrative":** Number of Administrative web pages visited by the shopper during the shopping session.
- **"Administrative Duration":** Amount of time spent on the Administrative web pages by the shopper during the shopping session.
- **"Informational":** Number of Informational web pages visited by the shopper during the shopping session.

- **"Informational Duration":** Amount of time spent on the Informational web pages by the shopper during the shopping session.
- **"Product Related":** Number of different types of web pages visited by the shopper during the shopping session.
- **"Product Related Duration":** Total time spent in each of the page categories from the "Product Related" feature.
- **"Bounce Rate", "Exit Rate" and "Page Value":** These represent the features used by Google Analytics for each page in the e-commerce site;

  Bounce Rate: percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session.

  Exit Rate: calculated as for all pageviews to the page, the percentage that were the last in the session.

  Page Value: average value for a web page that a user visited before completing an e-commerce transaction.

- **"Special Day":** This feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day, Christmas, Thanksgiving, etc.) in which the sessions are more likely to be finalized with transaction.
- **"Month":** Month of the year the shopping session happened.
- **"Operating System":** The type of operating system used by the shopper.
- **"Browser":** The type of browser used by the shopper.
- **"Region":** Region the shopper is located.
- **"TrafficType":** The type of traffic experiencedby the shopper during the shopping session.
- **"VisitorType":** What kind of visitor is the shopper; could be returning or new visitor.
- **"Weekend":** True/False feature indicating whether the session happened during the weekend or not.
- **"Revenue":** True/False feature indicating whether a transaction happened or not.

Of the 12,330 sessions in our dataset, 84.5% $(10,422)$ were negative class samples that did not end with a transaction (Revenue = FALSE), and the rest (1908) were positive class samples ending with a transaction (Revenue = TRUE). Our goal in this project is to discuss what factors influence a shopper's decision to complete a purchase the most. We believe that identifying these factors would help retailers stimulate revenues in the most efficient ways.

Therefore, "Revenue" would be our dependent variable. In our project, we want to start with some dimension reduction procedures to select a few most significant variables among our 17 independent features using both LASSO regression and FDR analysis.

After variable selection, we want to start our model analysis by selecting two models to fit into our data and evaluate how they perform: logistic regression (as our dependent variable is categorical), and a decision tree model.

1.3 Preliminary Analysis

In this section, we would like to take a first look at our 18 features and interpret their structure a little better before proceeding with our dimension reduction and variable selection procedures.

Administrative

The administrative feature represents the number of administrative websites the shopper looked at during our shopping session.

Here is a quick summary of the values of Administrative:

```
admin
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
5768 1354 1114  915  765  575  432  338  287  225  153  105   86   56   44   38   24   16   12    6
  20   21   22   23   24   26   27
   2    2    4    3    4    1    1
```

Figure 1.1: Administrative feature summary

For a more direct representation, here is a histogram:

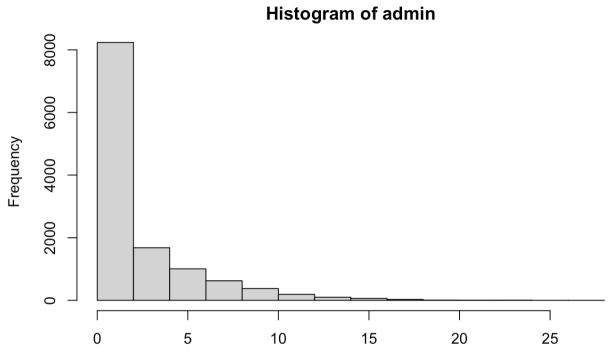

Figure 1.2: Administrative feature histogram

Both summaries demonstrated that the number of administrative websites visited by most shoppers are either zero or very small.

Administrative duration

The administrative duration feature represents the time spent on administrative websites by each shopper. Since this feature is continuous, we take a look at its statistical summary:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00    0.00    7.50   80.82   93.26 3398.75
```

Figure 1.3: Administrative Duration feature summary

A box-plot of this feature looks like:

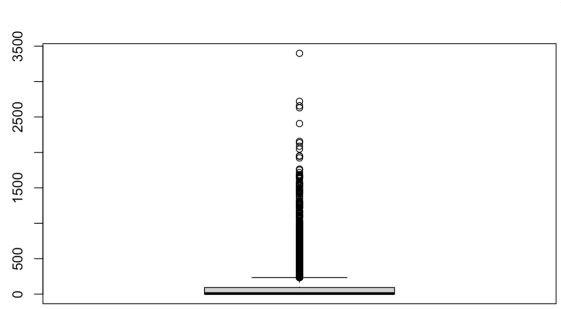

Figure 1.4: Administrative Duration boxplot

Like Administrative, this feature skews heavily toward zero; this follows from the fact that Administrative feature skews toward zero as well.

Informational

This feature represents the number of informational webpages visited by the shopper. Here is a summary of the values of Informational;

```
info
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   16   24
9699 1041  728  380  222   99   78   36   14   15    7    1    5    1    2    1    1
```

Figure 1.5: Informational summary

As well as a histogram to demonstrate its distribution:

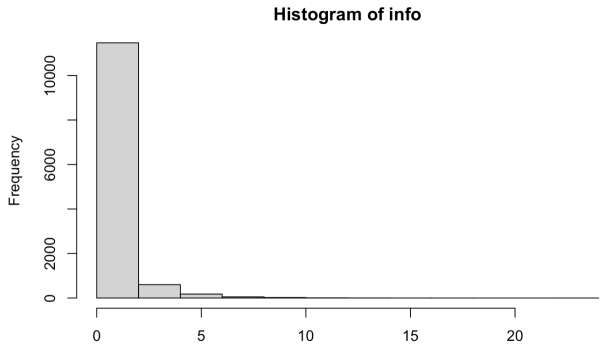

Figure 1.6: Informational histogram

This feature behaves very much like the Administrative feature, being very right-skewed and most values zero. In addition, the Informational feature is even more left-skewed than the Administrative feature.

Informational Duration

This feature represents the amount of time spent on informational websites during each session for each shopper. We take a look at its statistical summary and boxplot:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00    0.00    0.00   34.47    0.00 2549.38
```

Figure 1.7: Informational Duration Summary

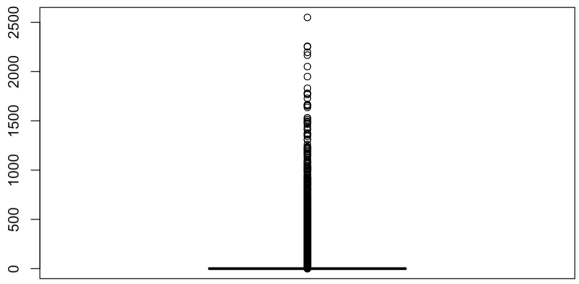

Figure 1.8: Informational Duration Boxplot

These visualizations confirm our expectations: Informational Duration is also very right-skewed and sparse, containing mostly zeros and small values.

ProductRelated

This feature represents the number of different types of web pages (related to product) visited by the shopper during the shopping session. We summarize this data with a boxplot:

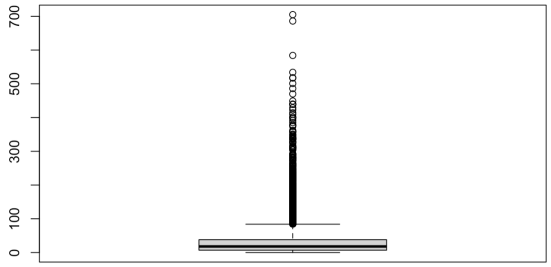

Figure 1.9: Product Related Boxplot

Again we see that most shoppers didn't look at product-related webpages or only looked at very few amounts of product-related webpages.

ProductRelated Duration

This feature captures the amount of time a shopper spends in the product-related webpage recorded in the previous feature. Let's take a look at a scatterplot of this feature.

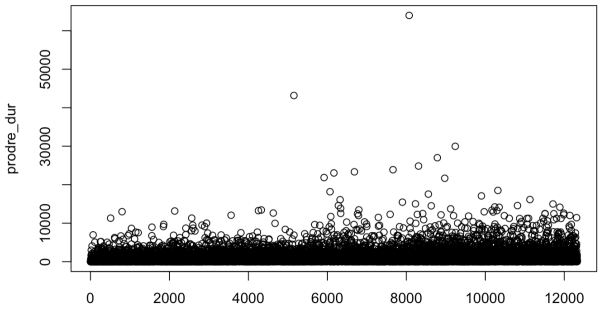

Figure 1.10: Product Related Duration Scatter-plot

We see the expected heavy concentration near zero as well as the clear outliers.

BounceRates

This feature represents the percentage of visitors who entered and left the webpage without triggering any other requests during the session. We look at a boxplot and a scatterplot:

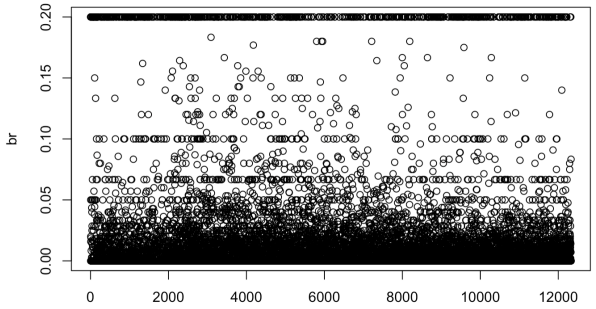

Figure 1.11: Bounce Rates Scatter-plot

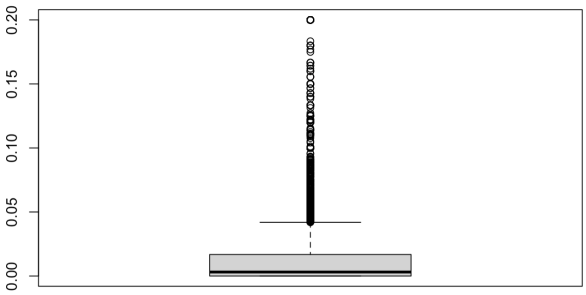

Figure 1.12: Bounce Rates boxplot

We see that there is a concentration around zero, but there are also significant distributions around 0.2 and even 0.1.

ExitRates

This feature represents for all pageviews to the page, the percentage that were last in the session. Like BounceRattes, we also take a look at the boxplot and scatterplot:

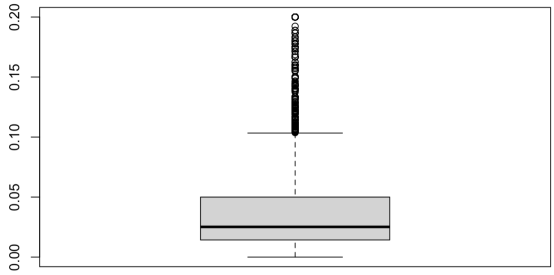

Figure 1.13: Exit Rates boxplot

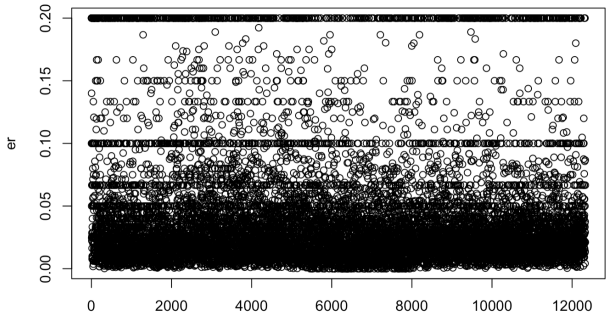

Figure 1.14: Exit Rates scatterplot

We see some signals around 0.1 and 0.2, therefore we could also do a histogram to see these signals more clearly.

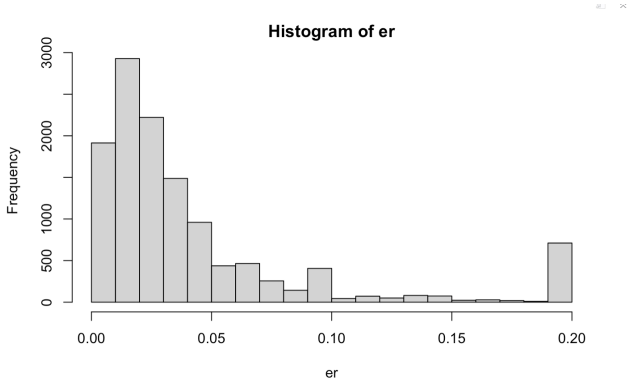

Figure 1.15: Exit Rates histogram

Compared to Bounce Rates, the Exit Rates are less concentrated near zero and more uniformly scattered. We also see concentrations around values 0.1 and 0.2.

PageValues

This feature records the average value for a web page that a user visited before completing an e-commerce transaction. We would only need a scatterplot for this feature:

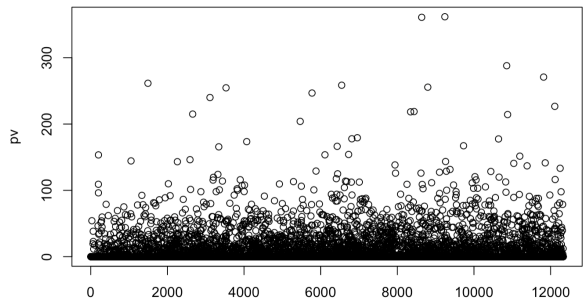

Figure 1.16: Page Values scatterplot

Again we see the majority page values are close to zero with a few outliers that may correspond to a higher transaction probability.

SpecialDay

This feature indicates how close the time of the transaction is to a special holiday. We would expect this feature to be highly correlated to the Revenue feature. Let's look at this histogram

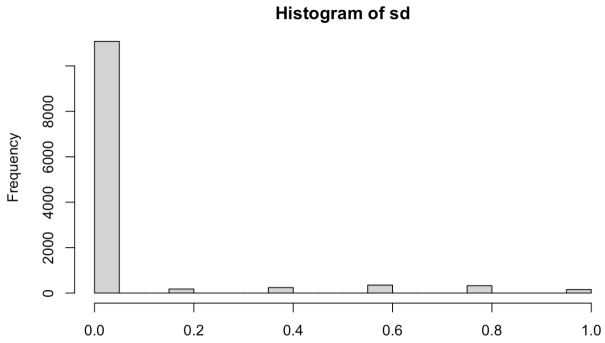

Figure 1.17: SpecialDay histogram

We see most shopping sessions happen nowhere near the special holidays, but there are a few that are very close to the holidays.

Month

Here is a summary of the months where the shopping sessions happen

```
month
 Aug  Dec  Feb  Jul June  Mar  May  Nov  Oct  Sep
 433 1727  184  432  288 1907 3364 2998  549  448
```

Figure 1.18: Month summary

We see significant increase in the number of shopping sessions in the months March, May, November, and December.

Weekend

This feature simply indicates whether the shopping session happened during the weekend or not. Let's look at a quick summary;

```
wk
FALSE  TRUE
 9462  2868
```

Figure 1.19: Weekend summary

While we might expect more shopping sessions to happen during the weekend, the weekend only consists of two days while the weekdays consist of five.

1.4 Dimension Reduction

1.4.1 Lasso Analysis

There are a total of 18 variables in the data set, which is relatively redundant to do predictions, so we decided to reduce the dimension of variables before attempting any model selections. The first technique we tried is using LASSO regression to choose significant variables.

Here below the result graph of lasso.

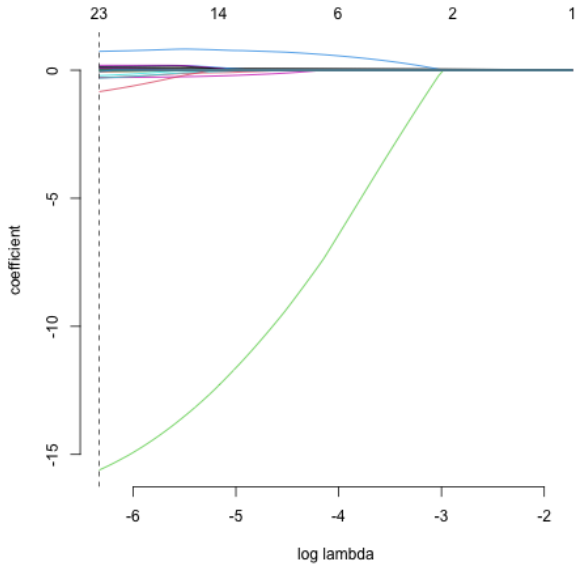

Figure 1.20: Lasso Graph

From the graph, we can see that except for one variable, all other variables behaves relatively similarly. With cross-validation, we chose the best λ to fit a best lasso:

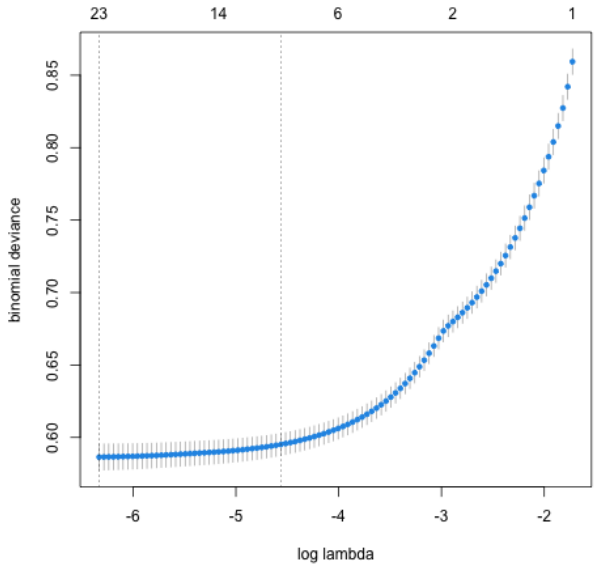

Figure 1.21: Lasso Graph

LASSO results were only able to help us eliminate three variables, not as many we would expect. (Please see detailed results in appendix A) Therefore, we tried an additional technique on reducing dimensions: FDR analysis.

1.4.2 FDR Analysis

We first ran a linear regression to get p-values of each variable (result in Appendix B), and plotted the distribution of p-values.

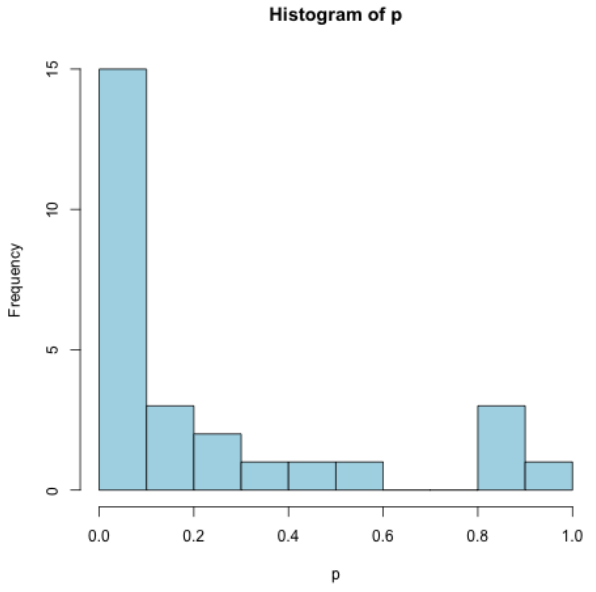

Figure 1.22: p-values Histogram

From the histogram of p-values, we can see that it is not a uniform distribution, and there is a spike around 0. Hence, it is necessary to run FDR analysis to pick significant variables. Then, we ran a couple of FDR analysis by different parameters of False Discovery Rate (q)

q = 0.1

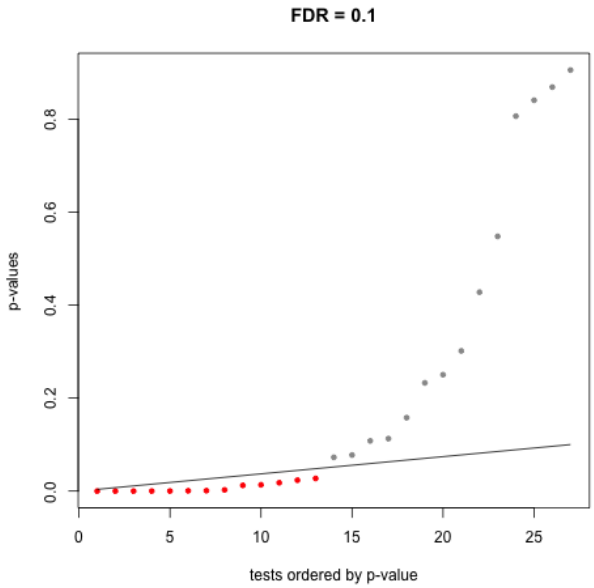

Figure 1.23: FDR (q = 0.1)

q = 0.05

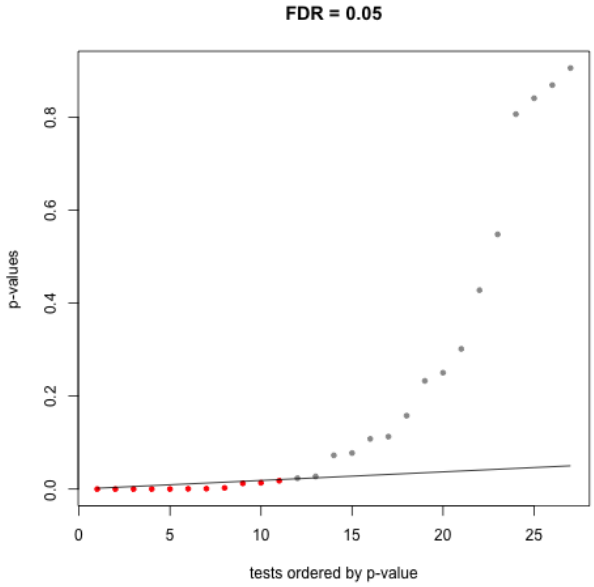

Figure 1.24: FDR (q = 0.05)

q = 0.01

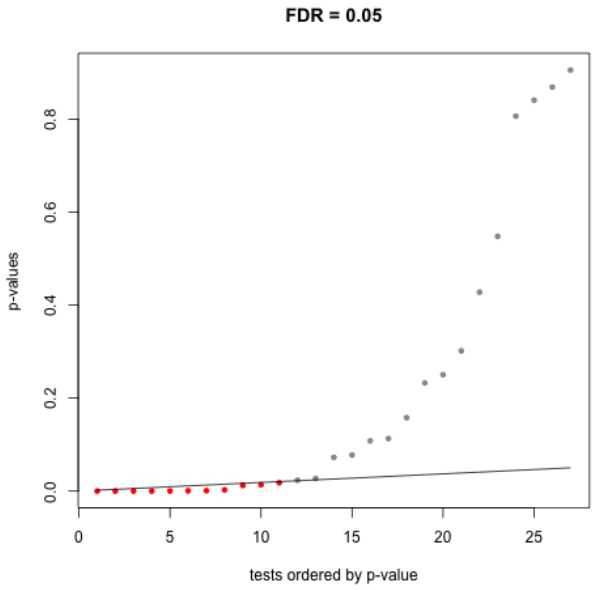

Figure 1.25: FDR (q = 0.01)

From these three figures, we believe $q = 0.01$ is most ideal, reducing a reasonable number of variables. $q = 0.1$ indeed reduced many variables, but there were still a large amount of significant variables, so we believe that it is reasonable to continue reducing the false discovery rate. The variables selected in the cases of $q = 0.05$ and $q = 0.01$ are the same, so we believe we have found the most suitable version of dimension reduction. If we continued to lower the value of q, we might eliminate some significant variables that might be essential to making predictions. Therefore, based on $q = 0.01$, we got the selected variables: ProductRelated_Duration, BounceRates, ExitRates, PageValues, MonthDec, MonthNov, VisitorTypeReturning_Visitor.

There are 10 months recorded in the original data, but based on the result of FDR, it shows that only December and November are significant, so we converted the original 'Month' variable to be a variable that only indicates whether it is December/November or not. If it is December/November, it is labeled as **TRUE**, otherwise it is labeled as **FALSE**.

Chapter 2

Model Analysis

2.1 Logistic Regression

The first natural selection of modeling is logistic regression, due to the classification nature of the dataset we are working with.

2.1.1 Vanilla Version

First, we run all selected covariates, which are 'ProductRelated_Duration', 'BounceRates', 'ExitRates', 'PageValues', 'VisitorType', 'Monthsig' and 'Revenue'. (from our earlier model selection)

The specific output is in the Appendix C. Here below is the deviance of the model.

```
Null deviance: 8433.7  on 9863  degrees of freedom
Residual deviance: 5774.6  on 9856  degrees of freedom
AIC: 5790.6
```

Based on the deviance, we calculated the R^2, which is

$$R^2 = 1 - (\frac{\text{Residual deviance}}{\text{Null deviance}}) = 1 - \frac{5774.6}{8433.7} = 0.3152946$$

R^2 shows that the model does not strongly capture the variation of the data. Therefore, we consider adding some interaction terms to explain more of data.

2.1.2 Interaction Version

We added 2 interaction terms, which are 'ExitRates*VisitorType' and 'ProductRelated_Duration*Monthsig'.

ExitRates*VisitorType'

'ExitRates' captures the percentage that the website visited was the last in the shopping; VisitorType takes two values: new visitor and returning visitor. In particular, for a new visitor, it is highly possible that one would visit every page website carefully, and then decide what they want to buy, which means that the average of each website is relatively equal, so there should be no big gap between the percentage

of the last website and those of other websites. If the customer is a returning visitor, one maybe directly goes to the website they want because they have the target. Recall that the histogram of 'ExitRates' and 'Administrative' have similar shapes: they are both right-skewed. ('Administrative' means the number of Administrative web pages visited by the shopper during the shopping session.) The less pages a customer visits, the more likely this customer is a returning visitor, the less time this visitor spend on other pages.

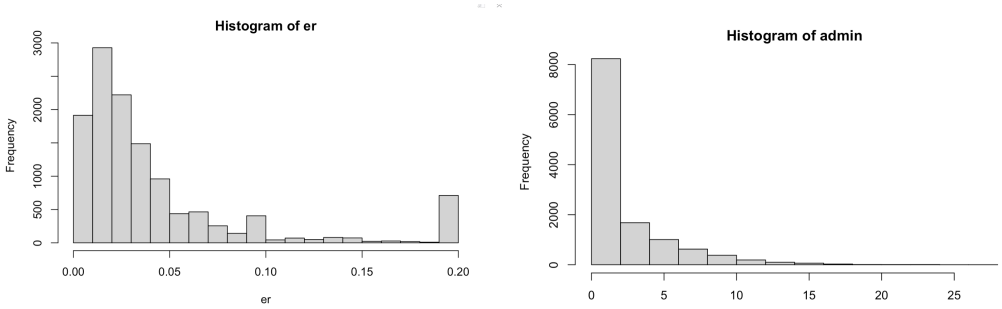

Figure 2.1: Exit Rates histogram

Consequently, these customers would not spend a lot time on other websites, so the average of the last one accounts for a large part of the total. Hence, we think these two features are related.

'ProductRelated_Duration*Monthsig'

Next, we think there is also a relation between duration and month: 'ProductRelated_Duration*Monthsig'. In the preliminary analysis and dimension reduction sections, we found that November and December are the significant months. From our point of view, the reason why they are more significant than other months is that there are many festivals and holidays in these two months, such as Christmas Day and Black Friday, which is similar to the covariate 'Special Day' in the original data set. During these days, there are a great number of discount events, resulting in customers buying much more products than at ordinary times of the year. These purchases usually connect with a long duration: customers do not need to buy them again in other months; so except for November and December, other months are not as significant. What's more, There are some special products to be purchased in November and December, which are not necessities during other times of the year, such as Christmas tree for Christmas Day and ski outfit for winter. Thus, we think November and December are shopping months for people, and people tend to purchase long-duration products.

Model Result

After adding two interaction terms, we run the logistic regression, and calculated R^2 again.

```
Null deviance: 8433.7  on 9863  degrees of freedom
Residual deviance: 5753.0  on 9853  degrees of freedom
AIC: 5775

Number of Fisher Scoring iterations: 7
```

However, we got $R^2 = 1 - \frac{5753.0}{8433.7} = 0.3178557$. It just increases 0.0025611, not an obvious improvement.

```
ExitRates:VisitorTypeOther                0.6366
ExitRates:VisitorTypeReturning_Visitor    0.0435 *
ProductRelated_Duration:MonthsigTRUE   2.83e-05 ***
```

Even though R^2 does not increases significantly, the new interaction terms showed significant effect, which validated what we guessed before. There is a significant relation between 'ExitRates' and 'Returning_Visitor'. Also, there is a significant relation between 'ProductRelated_Duration' and 'Monthsig'.

2.1.3 Prediction

Then, we used these two logistic models to do predictions with testing data set to calculate the OOS R^2.

- For the vanilla logistic regression, OOS $R^2 = 0.3412736$ with residual deviance = 221.1956 and null deviance = 335.7928.
- For the interaction verison, OOS $R^2 = 0.3461579$ with residual deviance = 219.5555 and null deviance = 335.7928.

From these two OOS R^2, we can see that there is no big difference between the in-sample R^2. Even though these two logistic model do not capture the variation very well, they are stable and are not subject to overfitting. Therefore, we think these two model are still useful guidelines for some dealers or website developers.

Meaningful Coefficient Interpretation

Since we still think these two models are useful guidelines, it is meaningful to interpret the coefficients of some significant variables, which gives a better understanding of the data set. Since the interaction version slightly increases R^2, we choose this one to interpret, and based on p-values, we choose two terms with the smallest p-values.

```
PageValues                             8.242e-02
ProductRelated_Duration:MonthsigTRUE   1.278e-04
```

1. 'PageValues': As page value calculated by Google Analytics increases one unit, the probability of a purchase on this website increases 8.59118%.
2. 'ProductRelated_Duration:MonthsigTRUE': If a customer visit the website at November or December, the probability of a purchase on this website increases 0.01278082%

2.2 Tree-Based Model

Another natural consideration when we are trying to decide what models to fit into our data is tree-based models. To start with, we want to fit a decision tree to our data, not only due to the easy-to-interpret nature

of decision trees, but also that we wish to predict a simple TRUE/FALSE decision (whether revenue is made or not), an objective that decision trees could help us achieve easily. However, even though we made several attempts and received interpretable results, we would ultimately decide that tree-based models are not the best option for our particular dataset, due to our dataset being rather high-dimensional.

2.2.1 Decision Tree

Simple Decision Tree

With the dataset after dimension reduction, we were able to fit a decision tree that looks like

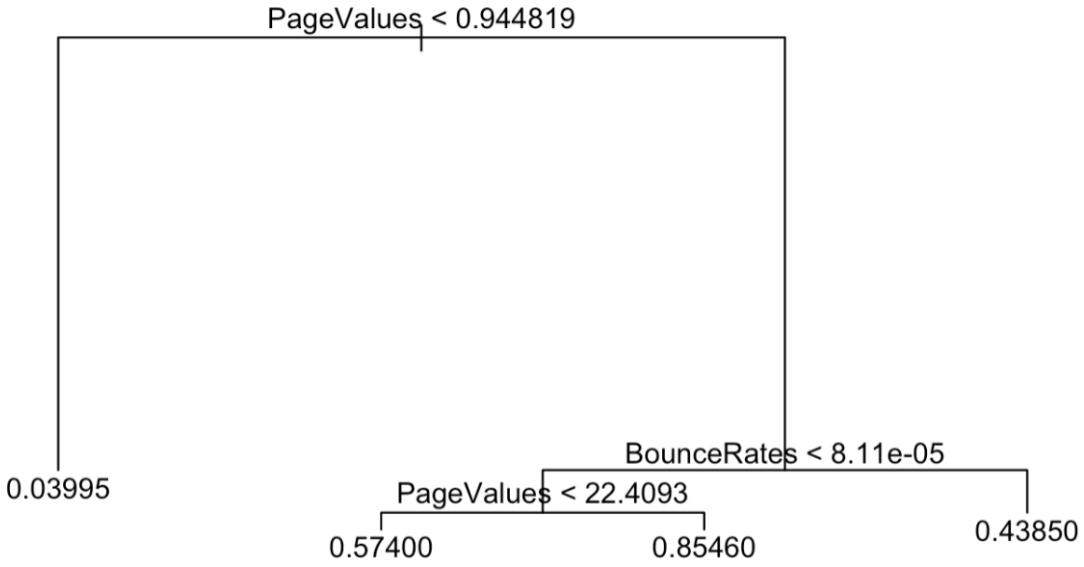

Figure 2.2: Decision Tree after Dimension Reduction

From this decision tree, we could make some reasonable interpretations. We interpret the leaf nodes as probabilities of obtaining a "TRUE" value for the "Revenue" outcome, i.e., the probability of actually making a purchase through the website. The model decided that the first decision to be made is whether the page value of our website is larger than or smaller than 0.944819. If the page value is smaller than 0.944819, then there is a very, very small chance that a purchase would be made: only 3.995%. The model is telling us that the feature page rank itself contains much information that could help us understand why some groups of website will have a very small chance of achieving revenue on average. This feature selection also makes sense as page rank is the relative importance of the website, or how big and famous the website might be. For instance, if a user search for sportswear, websites of brands such as Nike, Adidas, Under Armour, etc. would show up at the top of their search due to these famous brands having higher page ranks. If a user search for makeup and skincare products, websites of brands such as Sephora, MAC, NYX, etc. would show up due to them having higher page rank. Therefore, higher page rank means more customer flows, resulting

in higher changes of a purchase being made.

Secondly, the model selects feature Bounce Rate, which is defined earlier as "percentage of visitors who enter the site from that page and then leave("bounce") without triggering any other requests to the analytics server during that session". A "bounce" likely occurs when a user accidentally entered the website, or that the user sees that the website is entirely helpless and leaves rather quickly, without any other actions. We could understand that the model chose his feature next to rule out the websites that have many "accidental clicks" or have many users who, after having one quick look, decided that this website is rather not helpful. We see that the choice of this feature can be interpreted easily, as users who "bounce" will definitely not make a purchase. The model decided that if a website has bounce rate smaller than $8.11 \cdot 10^{-5}$, we would go further down the tree, and if a website has bounce rate larger than $8.11 \cdot 10^{-5}$, there will be a 43.85% chance that a revenue could be made.

The last step chosen by the model goes back to page rank, giving the website a 85.46% chance to be able to make revenue if the website has a page rank larger than 22.4093 (in addition to this website having a bounce rate less than $8.11 \cdot 10^{-5}$).

At first glance, this model makes intuitive sense, and it agrees with our earlier logistic regression by selecting the page rank as the most important feature. However, some of the details in this model might not be as interpretable as we expected. For instance, the threshold for the Bounce Rate results in the data being divided into two groups, among which two of the three resulting leaf nodes give probability of revenue rather close to 50%, which can be very difficult to categorize into TRUE or FALSE, as they lie very closely to our decision boundary. In addition, the model doesn't seem to incorporate more information from the dataset by focusing mostly on the role of the page rank.

Now we try to fit a decision tree on the entire dataset with out dimension reduction, and we have the following result

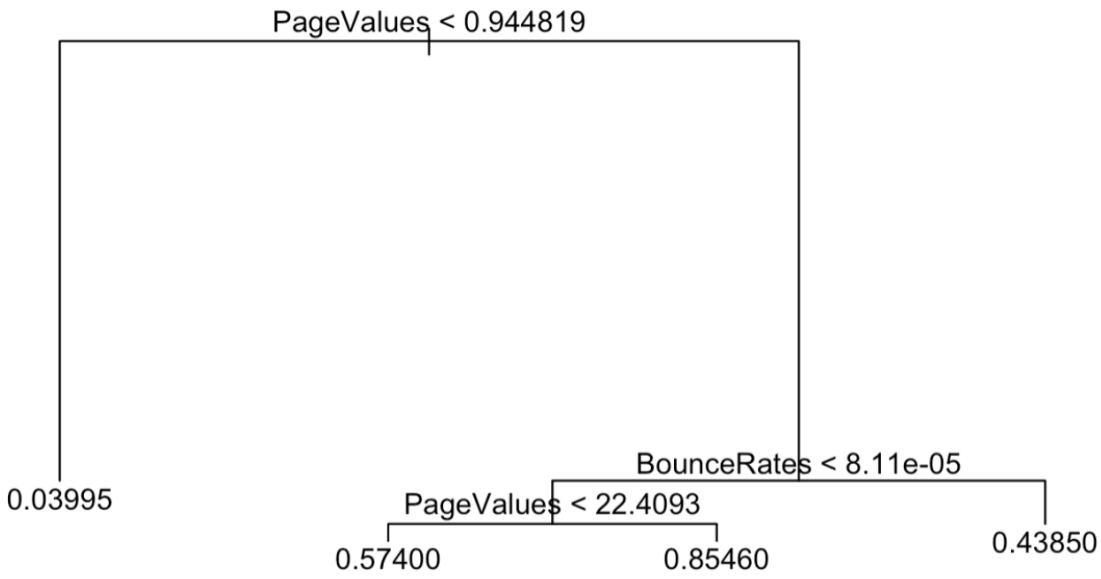

Figure 2.3: Decision Tree without Dimension Reduction

We see virtually no difference in the structure of the tree, and only minor adjustment in the parameters. This could probably be explained by that the features we left out in the dimension reduction don't contain much information for this particular model.

2.2.2 Random Forest

Lastly, we made an attempt to fit a random forest, hoping that a large number of relatively independent decision trees operating together will outperform any single decision tree. Here is the result:

```
Call:
 randomForest(formula = Revenue ~ ., data = TrainSet, importance = TRUE,
ntree = 100, cutoff = 2, mtry = 6)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 6

          Mean of squared residuals: 0.07200084
                    % Var explained: 43.79
```

Figure 2.4: Random Forest

We see that R chose to do a regression random forest on our data, and the resulting percentage of variance explained is around 43%. However, the random forest model was bad at prediction on our testing data (20% of the entire dataset, selected at random) as you can see in section E in the appendix: the prediction results

we got from random forest were not ideal at all. This might be the result of our very large dataset, containing over 10,000 sample points, making the random forest model very much overfitting, therefore performs poorly in out-of-sample predictions.

2.2.3 Conclusion of Tree-Based Models

In essence, even though we have valid motivations, our attempts at tree-based models, with or without bagging, are not successful. Due to the non-linear nature of tree-based models and the large amount of samples we have in the dataset, the tree-based models, even with bagging, is heavily affected by overfitting, and the accuracy for predictions was not ideal at all.

2.3 Naive Bayes Classifiers

Based on the Bayes Theorem, Naive Bayes Classifier is one of the most widely-used machine learning algorithms for classification. We decided to select this particular supervised learning technique not only because it works very well with classification tasks, but also because it uses a probabilistic approach and outputs are generated instantly, a particular advantage we favor in our large dataset setting.(5,)

2.3.1 Naive Bayes

Due to the result of the tree-based model, we did not learn on the data set with dimension reduction. Since this data set is a classification problem, the dimension reduction from regression might lead to errors. Hence, we kept all variables in the following analysis.

We started by splitting our dataset randomly into training and testing sets, with 80% of the data being in the training set and 20% in the testing set, and attempting to fit a simple Naive Bayes model with a 10-fold cross-validation. Here is a summary of the results:

```
Naive Bayes
9865 samples
  17 predictor
   2 classes: 'FALSE', 'TRUE'
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 8878, 8878, 8880, 8878, 8878, 8880, ...
Resampling results across tuning parameters:
  usekernel  Accuracy   Kappa
  FALSE      0.9558052  0.8449586
   TRUE      0.9912818  0.9662854
Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
```

```
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

We could see that the model successfully made classification between the two classes we desire to see: True or False for the "Revenue" feature. The accuracy on the training dataset is very high, around 95.5% for Revenue feature to be classified as False, and 99% for the Revenue feature to be classified as True.

In terms of prediction, we could use the model we just fitted onto our testing set, and produce a confusion matrix by comparing the predicted outcomes of Revenue feature and the true outcome of the Revenue feature:

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE TRUE
     FALSE  2079   17
     TRUE      5  364
               Accuracy : 0.9911
                 95% CI : (0.9865, 0.9944)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : < 2e-16
                  Kappa : 0.9654
 Mcnemar's Test P-Value : 0.01902
            Sensitivity : 0.9976
            Specificity : 0.9554
         Pos Pred Value : 0.9919
         Neg Pred Value : 0.9864
             Prevalence : 0.8454
         Detection Rate : 0.8434
   Detection Prevalence : 0.8503
      Balanced Accuracy : 0.9765
       'Positive' Class : FALSE
```

We see that in terms of our-of-sample predictions, our Naive Bayes model did really well by reaching an accuracy of 99.11%. Among all revenues where the true outcome is False, the model made correct predictions for 2079 out of 2084 cases, making only 5 mistakes. Among all revenues where the true outcome is True, the model made correct predictions for 364 out of 381 cases, making only 17 mistakes. Also, 'P-Value [Acc $>$ NIR] : 2e-16' shows that the accuracy is significantly better than "no information rate", and this p-value is extremely small. In essence, the Naive Bayes model performed very well on our dataset.

In addition, here is a summary of the prior distributions used in this model. To see the detailed prior distributions for each feature, please refer to appendix F, where some visualizations are also included.

```
============================== Naive Bayes ==============================


- Call: naive_bayes.default(x = X_gnb, y = y_gnb)
- Laplace: 0
- Classes: 2
- Samples: 9865
- Features: 17
- Conditional distributions:
    - Bernoulli: 1
    - Categorical: 2
    - Gaussian: 14
- Prior probabilities:
    - FALSE: 0.8452
    - TRUE: 0.1548
```

2.3.2 Gaussian Naive Bayes

Since our data describes attributes of online shopping, a very common part of our daily life, we decided that it would be reasonable to assume a Gaussian prior distribution and expand our modeling options by attempting to fit a Gaussian Naive Bayes classifier. Even though in the preliminary section, the histogram of many features are left-skewed, instead of having an exact normal distribution shape, most of them still have bell-shaped distributions with a spike. We believe with enough data, the assumption for normal distribution is valid. To fit the Gaussian Naive Bayes classifier, we start by separating 80% of data to be the training data set, and setting the Revenue feature as the response variable y, which is binary.

```
=================== Gaussian Naive Bayes ====================


- Call: gaussian_naive_bayes(x = X_gnb, y = y_gnb)
- Samples: 9865
- Features: 17
- Prior probabilities:
    - FALSE: 0.8452
    - TRUE: 0.1548
-------------------------------------------------------------
```

From the result of training data, the prior for making a purchase online is 0.1548, while the prior for not making a purchase is 0.8452, which is much higher.

We then applied this model to our testing dataset and evaluated the predictions by taking a look at the confusion matrix:

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE TRUE
     FALSE  1766  127
     TRUE    318  254
               Accuracy : 0.8195
                 95% CI : (0.8037, 0.8345)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : 0.9998
                  Kappa : 0.4267
 Mcnemar's Test P-Value : <2e-16
             Sensitivity : 0.8474
            Specificity : 0.6667
         Pos Pred Value : 0.9329
         Neg Pred Value : 0.4441
             Prevalence : 0.8454
         Detection Rate : 0.7164
   Detection Prevalence : 0.7680
      Balanced Accuracy : 0.7570
       'Positive' Class : FALSE
```

From the confusion matrix, we see that the accuracy is 81.95%, which is lower than the accuracy that general Naive Bayes got. Also, we observed that 'P-Value [Acc $>$ NIR] : 0.9998', which indicates that there is no significant improvement of accuracy, comparing with "no information rate". Thus, even though the accuracy is relatively high, Gaussian Naive Bayes classifier does not work perfectly. Moreover, compared with p-value of general Naive Bayes, 2e-16, it shows that the general case is better than the Gaussian case.

2.3.3 Conclusion of Naive Bayes

Consequently, the general Naive Bayes classifier is better. It won in both accuracy and p-value of "no information rate". Our previous guess that this data set fits normal distribution better did not match with the Gaussian Naive Bayes classifier.

2.4 Kernel Methods

Another common supervised classification model is the support vector machine (SVM). It is an algorithm that finds the hyper-plane that differentiates our classes (4,). While typical learning algorithms would accomplish this task by learning the most common features that best differentiate one class from another class and basing the classification decisions on those representative characteristics, SVM instead looks for

most similar sample data points between classes, which are called "supporting vectors". In other words, typical learning algorithms focus on the difference between classes, while SVM learns the similarities between different classes, and finds the best hyperplane separating the classes based on the most similar data points, or support vectors(1,). There are many category variables in our data set. When they are converted into dummies variables, they all consist of numbers like 0,1,2. Then, it leads to a lot of similarities between each variable. Furthermore, from previous results of lasso and decision tree, we can see that it is hard to figure out the difference between of each variable. Hence, we think it is a good choice to do SVM on this data set, since SVM focuses on learning similarities.

In addition, we also considered using a kernel trick on SVM. Kernel trick is to make a transformation between linearity and non-linearity, so adding kernel might be helpful for us to learn variables deeply. What's more, adding kernels would help us represent prior probability distributions of the data, instead of simple enumerations. If we cannot make any assumptions of normality, a kernel estimator would help produce better predictions given its representation of the data is correct.

There are three common kernels.

- Linear Kernel 'vanilladot': $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
- RBF Kernel 'rbfdot': $k(\mathbf{x}, \mathbf{x}') = \exp(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$
- Polynomial Kernel 'polydot': $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T\mathbf{x}')^d$

  where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$

Since our data set is not a particularly complicated one, we think these three kernels are already enough as candidates. Linear kernel is useful when the data set is large sparse matrix(2,). There are many categorical variables with 0 as entries, so we think linear kernel might be a valid option. In addition, as stated before, we believe that with enough samples, it's highly possible for our data to follow a normal distribution, so RBF Kernel might be a good choice, too.

2.4.1 RBF Kernel

Here we have the result of SVM with RBF kernel. Except for the parameter of cross-validation, we used the default parameter for other arguments. Based on the experiments of previous several models, we think complex parameters would leads to inaccurate results and high runtime complexity. We set the number of cross validation to be 10.

```
Support Vector Machine object of class "ksvm"
SV type: C-svc  (classification)
 parameter : cost C = 1
Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0768448509306368
Number of Support Vectors : 2798
```

```
Objective Function Value : -2070.64
Training error : 0.090421
Cross validation error : 0.106741
Probability model included.
```

From the training error, 0.090421, and cross validation error, 0.106741, we can see that the model learned well.

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE TRUE
     FALSE  2017  200
     TRUE     67  181
               Accuracy : 0.8917
                 95% CI : (0.8787, 0.9037)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : 1.855e-11
                  Kappa : 0.5166
 Mcnemar's Test P-Value : 6.569e-16
            Sensitivity : 0.9679
            Specificity : 0.4751
         Pos Pred Value : 0.9098
         Neg Pred Value : 0.7298
             Prevalence : 0.8454
         Detection Rate : 0.8183
   Detection Prevalence : 0.8994
      Balanced Accuracy : 0.7215
       'Positive' Class : FALSE
```

The above table is a summary of predictions on test data set. The accuracy is 0.8971, which is a good result. It is not a sign of overfitting nor a sign of under-fitting. Also, 'P-Value [Acc $>$ NIR] : 1.855e-11', is almost 0, indicating that the accuracy is significantly better than "no information rate". We think the SVM model with RBF kernel fits well.

2.4.2 Linear Kernel

We also made an attempt at a linear kernel: similar to what we did before, aside from the parameter of cross validation, we did not set any other hyper-parameter.

```
Setting default kernel parameters
Support Vector Machine object of class "ksvm"
```

```
SV type: C-svc  (classification)
 parameter : cost C = 1
Linear (vanilla) kernel function.
Number of Support Vectors : 2404
Objective Function Value : -2377.263
Training error : 0.113938
Cross validation error : 0.113838
```

Training error is 0.113938 and cross validation error is 0.113838. They are both higher than RBF kernel.

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE TRUE
     FALSE  2030  220
     TRUE     54  161
               Accuracy : 0.8888
                 95% CI : (0.8758, 0.901)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : 3.068e-10
                  Kappa : 0.4826
 Mcnemar's Test P-Value : < 2.2e-16
            Sensitivity : 0.9741
            Specificity : 0.4226
         Pos Pred Value : 0.9022
         Neg Pred Value : 0.7488
             Prevalence : 0.8454
         Detection Rate : 0.8235
   Detection Prevalence : 0.9128
      Balanced Accuracy : 0.6983
       'Positive' Class : FALSE
```

The accuracy from the confusion matrix is 0.8888, which is higher than that of RBF kernel as well. Thus, we could conclude that RBF kernel is better than linear kernel, even though the data set is a large sparse matrix.

2.4.3 Polynomial Kernel

We did not perform polynomial kernel successfully because the cost was too large. It took more than 30 minutes to learn the model and we decided that the high runtime cost rendered this model not a good fit for our dataset.

2.4.4 Conclusion of Kernel Methods

Therefore, based on three different kernels, SVM with RBF kernel performed best and cost least. They reached 0.8917 and 0.8888 test accuracy respectively, which means that they are likely not affected by overfitting. We think these models could produce a marked effect in reality.

Chapter 3

Conclusion

On this data set, we performed 4 kinds of techniques with 8 different models in total. They are logistic regression, tree-based model, Naive Bayes Classifier and Kernelized SVM.

Even though this is a classification problem, the logistic model with dimension reduction did worst, it only achieved around 30% R^2. Adding interaction terms also did not improve the model in an obvious way.

In terms of tree-based models, we attempted decision trees and random forests. However, this data set did not fit tree-based model well. At a first glance, the results of decision tree looked good, but they did not make sense in details. Also, it seems that decision tree did not incorporate information from all features in the data set, no matter we run it on the data set with dimension reduction or without. The result of decision tree only focused on two variables, 'Page Value' and 'BounceRates'. Random fores did even worse than decision tree, accumulating only approximately 20% accuracy.

The third kind technique we tried was Naive Bayes classifier. We first tried the general version of Naive Bayes, which means we did not assume any distribution for the data set. The result is good with 0.9911 test accuracy, which is extremely high. Then, we made an normal distribution assumption on the data set and tried Gaussian Naive Bayes classifier. The result is also not bad, compared with previous two techniques, logistic regression and tree-based model, but GNB has a lower test accuracy than the general version.

The final model we tried is kernerlized SVM. We considered the advantages of SVM and kernel tricks and chose three different kernels to analyze the data. We chose RBF kernel, linear kernel and polynomial kernel. SVM with RBF kernel performed best, while the polynomial kernel cost most and did not succeed.

To summarize, from the perspective of prediction, Naive Bayes performs best and takes good advantages of the whole data set.

Chapter 4

Discussion

In this project, we explored the factors that make an impact on online shopper's decision. We run a total of eight models, and reached some rather surprising results. Before we started our analysis, we assumed that the logistic regression and decision tree would fit the model perfectly, since this is a classification problem and there are enough variables. However, from the results of logistic regression and decision tree, these two did not reach what we expected at the beginning. Surprisingly, Naive Bayes classifier worked the best. This is a simple model. The reason why, we think, NB works best is that this data is not so complicated. It seems that the factor which allows people to decide to purchase is only whether this merchandise is worthy, which is similar to the meaning of page value. Other factors does not play too much a role. For example, the time that people leave in certain page does not show much information about their decision. It is possible that they just want to have a look. It is a highly personal motivation.

In terms of predictions, we finally concluded that Naive Bayes classifier gives the best results. The test accuracy is around 99%, which seems good. However, from the perspective of machine learning, we still think it has a potential to be overfitting. The training and test data set are both separated from the same data set. In other words, there might be bias in the collection of data. As we discussed in the first paragraph, there are many highly personal, subjective motivations involved. It is likely that the results may change as we collect the data in a totally different environment. In those cases, our models may have less reference values. Hence, we think there is some risk on models with such perfect test accuracy.

In conclusion, even though this data set is a little bit special, and there are some non-ideal results of the model, we still think our models could give a good reference to website developers and online dealers.

Chapter 5

Appendix

5.1 A

```
28 x 1 sparse Matrix of class "dgCMatrix"
                                        s0
(Intercept)                   1.438819e-01
(Intercept)                   .
Administrative                1.653580e-03
Administrative_Duration       .
Informational                 2.788054e-03
Informational_Duration        8.911197e-06
ProductRelated                2.936935e-04
ProductRelated_Duration       8.916093e-06
BounceRates                   5.961999e-02
ExitRates                    -6.156807e-01
PageValues                    8.916853e-03
SpecialDay                   -1.470737e-02
MonthDec                     -2.730748e-02
MonthFeb                     -2.936057e-02
MonthJul                      1.172393e-02
MonthJune                    -3.910305e-03
MonthMar                     -1.626034e-02
MonthMay                     -1.931373e-02
MonthNov                      7.867539e-02
MonthOct                      1.482433e-02
MonthSep                      5.898464e-03
OperatingSystems             -6.073449e-03
```

```
Browser                          1.984658e-03
Region                          -1.065718e-03
TrafficType                      .
VisitorTypeOther                -4.726952e-02
VisitorTypeReturning_Visitor -4.498561e-02
WeekendTRUE                      4.548500e-03
```

5.2 B

```
Call: lm(formula = Revenue ~ ., data = shoppers)

Residuals:
     Min       1Q   Median       3Q      Max
-2.46459 -0.14245 -0.07389 -0.00898  0.99417

Coefficients:
                               Estimate Std. Error t value
(Intercept)                    1.759e-01  1.881e-02   9.355
Administrative                 1.864e-03  1.159e-03   1.608
Administrative_Duration       -1.605e-05  2.025e-05  -0.793
Informational                  3.374e-03  2.935e-03   1.150
Informational_Duration         1.523e-05  2.534e-05   0.601
ProductRelated                 2.317e-04  1.312e-04   1.766
ProductRelated_Duration        1.030e-05  2.998e-06   3.434
BounceRates                    4.768e-01  1.437e-01   3.318
ExitRates                     -1.016e+00  1.515e-01  -6.707
PageValues                     8.944e-03  1.524e-04  58.683
SpecialDay                    -1.650e-02  1.596e-02  -1.033
MonthDec                      -4.996e-02  1.650e-02  -3.028
MonthFeb                      -6.034e-02  2.731e-02  -2.210
MonthJul                       2.468e-03  2.079e-02   0.119
MonthJune                     -3.289e-02  2.328e-02  -1.413
MonthMar                      -3.872e-02  1.635e-02  -2.369
MonthMay                      -4.011e-02  1.601e-02  -2.505
MonthNov                       6.244e-02  1.577e-02   3.960
MonthOct                       3.246e-03  1.967e-02   0.165
MonthSep                      -5.051e-03  2.061e-02  -0.245
OperatingSystems              -8.074e-03  3.269e-03  -2.470
```

```
Browser                         3.886e-03  1.713e-03   2.268
Region                         -1.841e-03  1.161e-03  -1.586
TrafficType                    -1.434e-04  7.133e-04  -0.201
VisitorTypeOther               -6.857e-02  3.819e-02  -1.795
VisitorTypeReturning_Visitor -4.811e-02  8.432e-03  -5.705
WeekendTRUE                     7.818e-03  6.551e-03   1.193

                               Pr(>|t|)
(Intercept)                     < 2e-16 ***
Administrative                 0.107947
Administrative_Duration        0.427871
Informational                  0.250257
Informational_Duration         0.547848
ProductRelated                 0.077487 .
ProductRelated_Duration        0.000596 ***
BounceRates                    0.000908 ***
ExitRates                      2.07e-11 ***
PageValues                      < 2e-16 ***
SpecialDay                     0.301497
MonthDec                       0.002470 **
MonthFeb                       0.027143 *
MonthJul                       0.905531
MonthJune                      0.157816
MonthMar                       0.017867 *
MonthMay                       0.012258 *
MonthNov                       7.53e-05 ***
MonthOct                       0.868927
MonthSep                       0.806414
OperatingSystems               0.013530 *
Browser                        0.023360 *
Region                         0.112797
TrafficType                    0.840665
VisitorTypeOther               0.072618 .
VisitorTypeReturning_Visitor 1.19e-08 ***
WeekendTRUE                    0.232735

---
Signif. codes:
```

```
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3055 on 12303 degrees of freedom
Multiple R-squared:  0.2881,Adjusted R-squared:  0.2866
F-statistic: 191.5 on 26 and 12303 DF,  p-value: < 2.2e-16
```

5.3 C

```
Call:
glm(formula = Revenue ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.1359  -0.4732  -0.3582  -0.1684   3.5167

Coefficients:
                               Estimate Std. Error z value
(Intercept)                  -1.965e+00  9.981e-02 -19.688
ProductRelated_Duration       1.187e-04  1.514e-05   7.841
BounceRates                  -8.231e-02  3.622e+00  -0.023
ExitRates                    -1.910e+01  2.702e+00  -7.068
PageValues                    8.223e-02  2.676e-03  30.729
VisitorTypeOther             -1.088e+00  6.009e-01  -1.811
VisitorTypeReturning_Visitor -3.801e-01  9.375e-02  -4.054
MonthsigTRUE                  5.944e-01  7.032e-02   8.452
                              Pr(>|z|)
(Intercept)                    < 2e-16 ***
ProductRelated_Duration      4.48e-15 ***
BounceRates                    0.9819
ExitRates                    1.57e-12 ***
PageValues                     < 2e-16 ***
VisitorTypeOther               0.0701 .
VisitorTypeReturning_Visitor 5.03e-05 ***
MonthsigTRUE                   < 2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8433.7  on 9863  degrees of freedom
Residual deviance: 5774.6  on 9856  degrees of freedom
AIC: 5790.6

Number of Fisher Scoring iterations: 7
```

5.4 D

```
Call:
glm(formula = Revenue ~ . + ExitRates * VisitorType + ProductRelated_Duration *
    Monthsig, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.1669  -0.4677  -0.3622  -0.1693   3.5187

Coefficients:
                                       Estimate
(Intercept)                          -1.984e+00
ProductRelated_Duration               4.359e-05
BounceRates                           8.274e-01
ExitRates                            -1.060e+01
PageValues                            8.242e-02
VisitorTypeOther                     -8.218e-01
VisitorTypeReturning_Visitor         -2.218e-01
MonthsigTRUE                          3.664e-01
ExitRates:VisitorTypeOther           -1.202e+01
ExitRates:VisitorTypeReturning_Visitor -1.014e+01
ProductRelated_Duration:MonthsigTRUE  1.278e-04
                                     Std. Error z value
(Intercept)                           1.224e-01 -16.206
ProductRelated_Duration               2.390e-05   1.824
BounceRates                           3.582e+00   0.231
ExitRates                             4.932e+00  -2.149
```

```
PageValues                                 2.679e-03  30.765
VisitorTypeOther                           7.988e-01  -1.029
VisitorTypeReturning_Visitor               1.288e-01  -1.722
MonthsigTRUE                               8.774e-02   4.176
ExitRates:VisitorTypeOther                 2.545e+01  -0.472
ExitRates:VisitorTypeReturning_Visitor     5.025e+00  -2.019
ProductRelated_Duration:MonthsigTRUE       3.052e-05   4.187

                                          Pr(>|z|)
(Intercept)                                < 2e-16 ***
ProductRelated_Duration                     0.0682 .
BounceRates                                 0.8173
ExitRates                                   0.0316 *
PageValues                                 < 2e-16 ***
VisitorTypeOther                            0.3036
VisitorTypeReturning_Visitor                0.0851 .
MonthsigTRUE                               2.97e-05 ***
ExitRates:VisitorTypeOther                  0.6366
ExitRates:VisitorTypeReturning_Visitor      0.0435 *
ProductRelated_Duration:MonthsigTRUE       2.83e-05 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8433.7  on 9863  degrees of freedom
Residual deviance: 5753.0  on 9853  degrees of freedom
AIC: 5775

Number of Fisher Scoring iterations: 7
```

5.5 E

```
mypred                  FALSE TRUE
  -1.83186799063151e-15     1    0
  -1.80411241501588e-15     3    0
  -1.77635683940025e-15     4    0
```

```
-1.74860126378462e-15    10    0
-1.72084568816899e-15    12    0
-1.69309011255336e-15    19    0
-1.66533453693773e-15    20    0
-1.63757896132211e-15    37    0
-1.60982338570648e-15    26    0
-1.58206781009085e-15    44    0
-1.55431223447522e-15    43    0
-1.52655665885959e-15    50    0
-1.49880108324396e-15    60    0
-1.47104550762833e-15    68    0
-1.4432899320127e-15     54    0
-1.41553435639707e-15    71    0
-1.38777878078145e-15    78    0
-1.36002320516582e-15    67    0
-1.33226762955019e-15    70    0
-1.30451205393456e-15    85    0
-1.27675647831893e-15    78    0
-1.2490009027033e-15    102    0
-1.22124532708767e-15    77    0
-1.19348975147204e-15    77    0
-1.16573417585641e-15    98    1
-1.13797860024079e-15    93    0
-1.11022302462516e-15    82    0
-1.08246744900953e-15    92    0
-1.0547118733939e-15      86    0
-1.02695629777827e-15    87    0
-9.99200722162641e-16    89    0
-9.71445146547012e-16    96    0
-9.43689570931383e-16    85    0
-9.15933995315754e-16    80    1
-8.88178419700125e-16    91    1
-8.60422844084496e-16    72    0
-8.32667268468867e-16    80    0
-8.04911692853238e-16    60    1
-7.7715611723761e-16      96    0
-7.49400541621981e-16 sig    0
```

5.6 F

This section includes the prior distribution used by our Naive Bayes model from section 2.3.1. (Features OperatingSystems, Browser, Region, TrafficType, and VisitorType are omitted):

```
-----------------------------------------------------
::: Administrative (Gaussian)
-----------------------------------------------------

Administrative    FALSE     TRUE
        mean    2.129168 3.408644
        sd      3.205487 3.713971
```

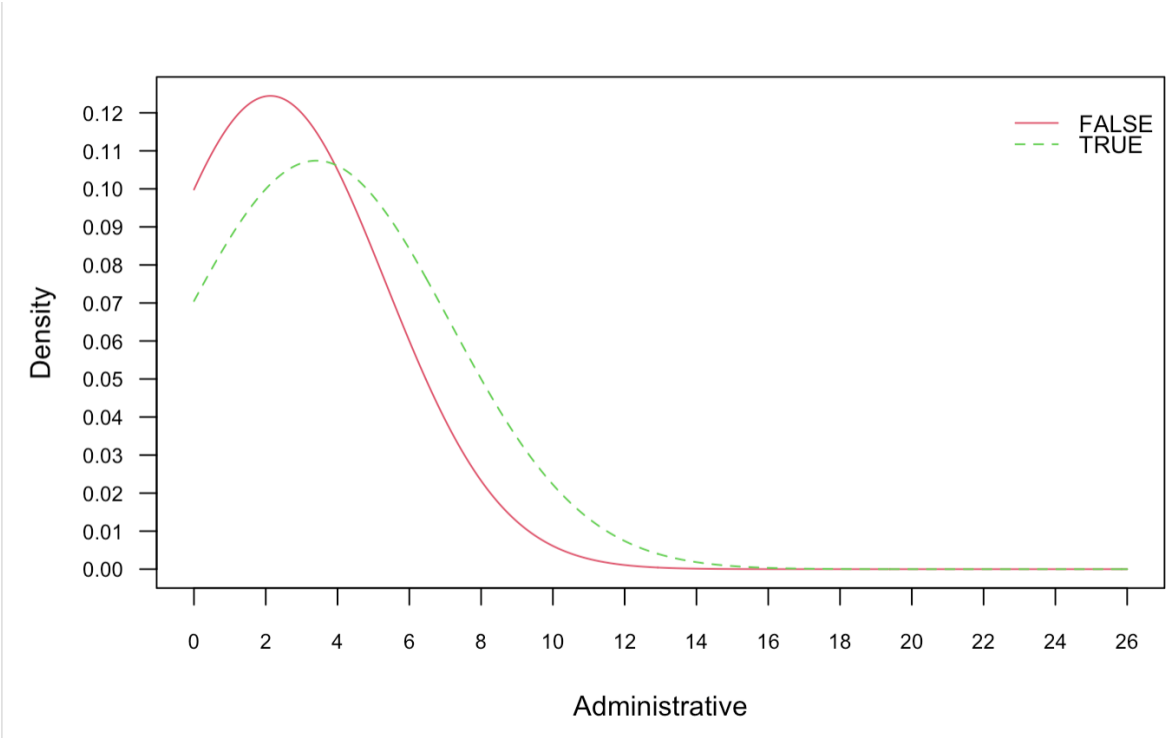

Figure 5.1: prior of feature "Administrative"

```
-------------------------------------------------
 ::: Administrative_Duration (Gaussian)
-------------------------------------------------

Administrative_Duration     FALSE      TRUE
                   mean  73.51576 118.99818
                   sd   168.58372 194.31793
```

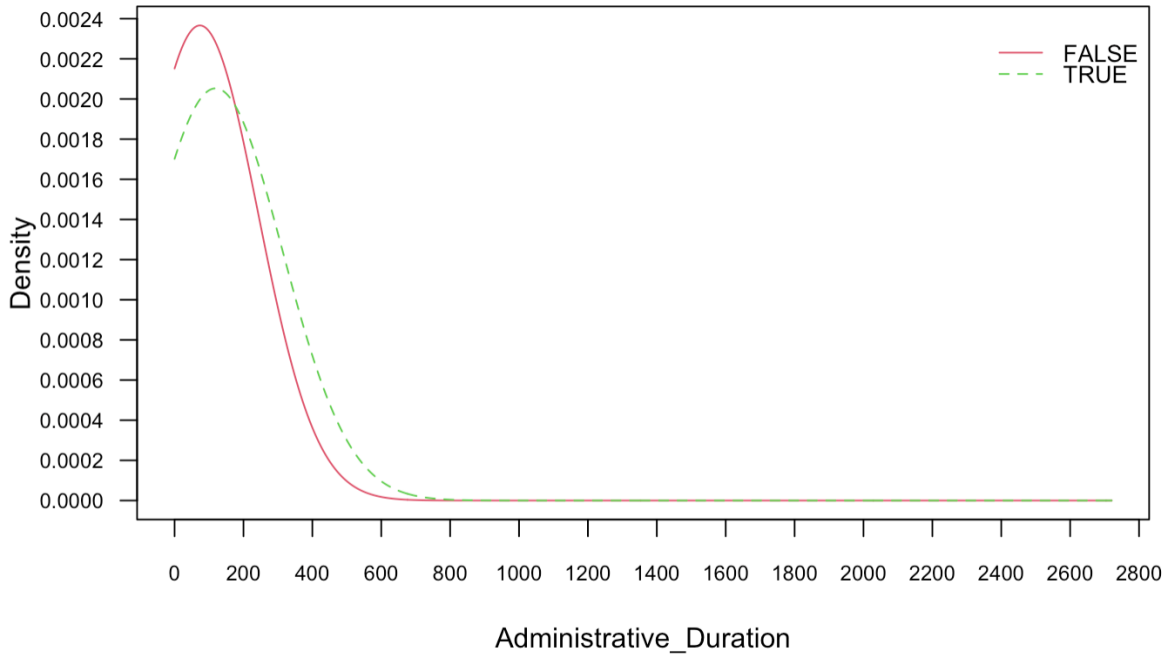

Figure 5.2: prior of feature "Administrative$_{D}uration$"

```
------------------------------------------------
 ::: Informational (Gaussian)
------------------------------------------------

Informational      FALSE      TRUE
         mean 0.4472296 0.7989522
         sd   1.1843043 1.5514924
```

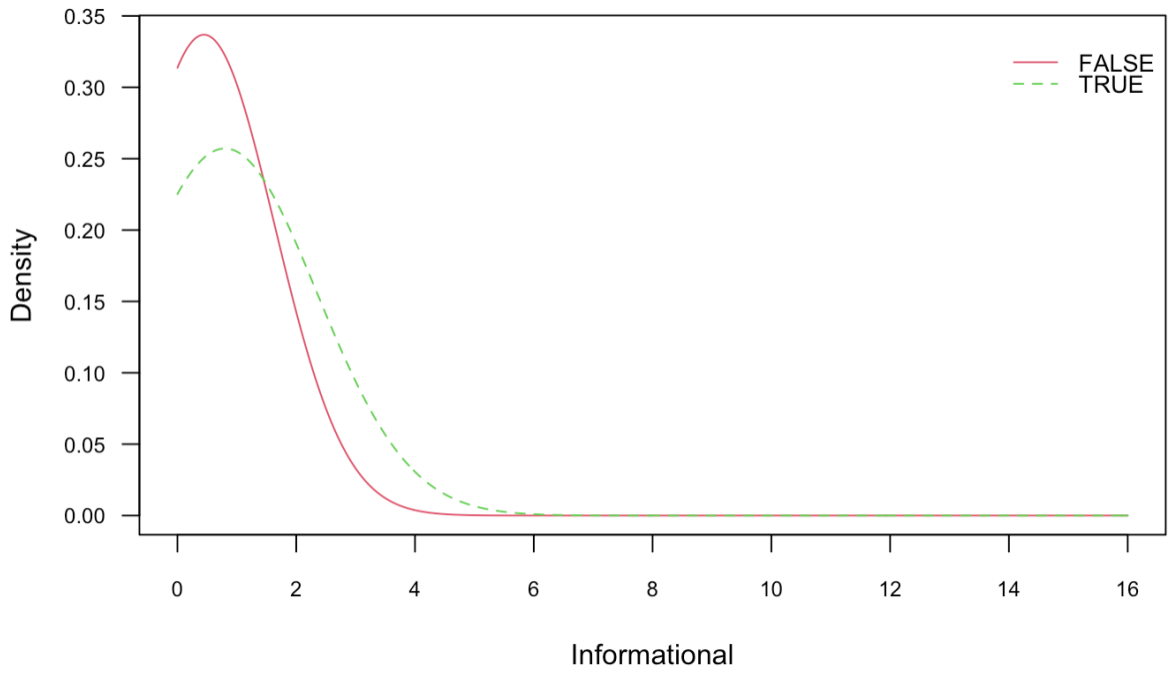

Figure 5.3: prior of feature "Informational"

```
------------------------------------------------
 ::: Informational_Duration (Gaussian)
------------------------------------------------

Informational_Duration     FALSE      TRUE
                  mean  29.41239  55.63147
                  sd   130.19956 166.08200
```

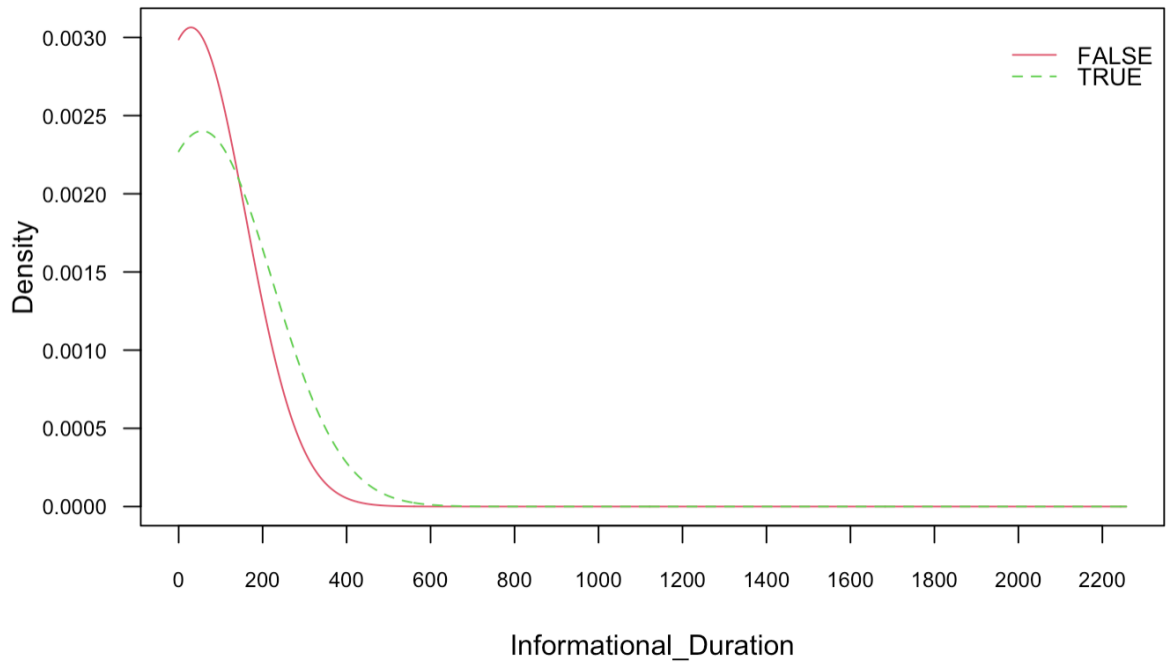

Figure 5.4: prior of feature "Informational$_Duration$"

```
------------------------------------------------
 ::: ProductRelated (Gaussian)
------------------------------------------------

ProductRelated    FALSE     TRUE
          mean 28.78916 48.49312
          sd   40.14687 58.62353
```

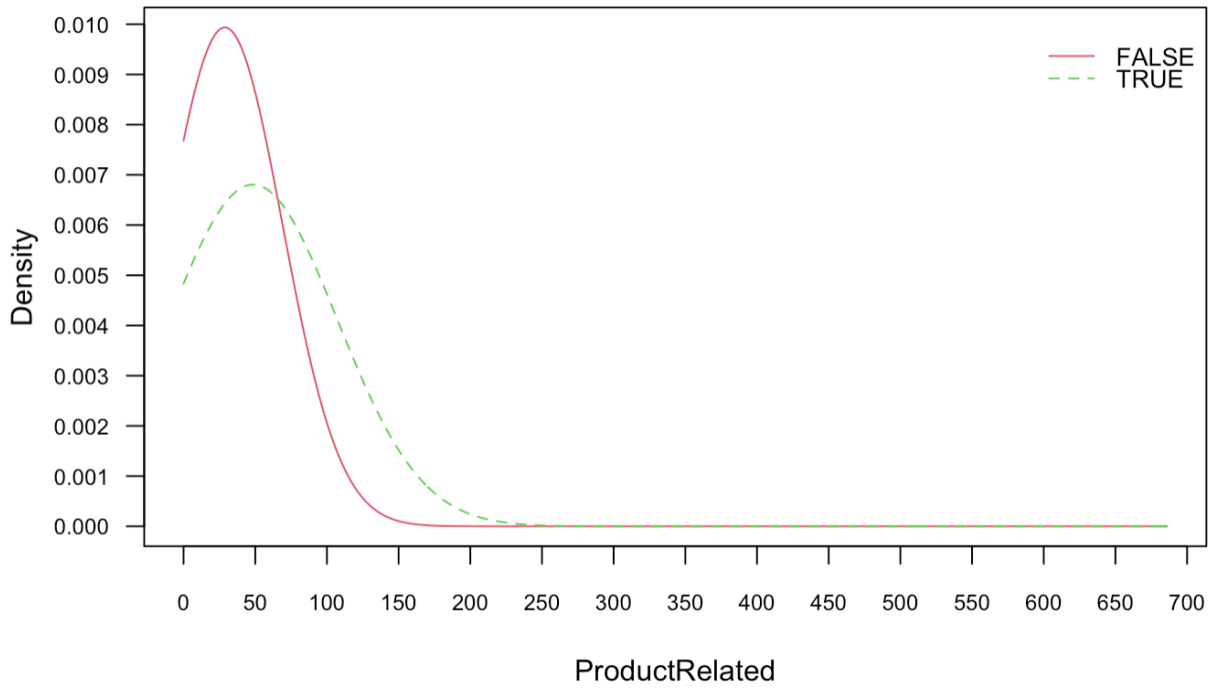

Figure 5.5: prior of feature "ProductRelated"

```
-------------------------------------------------
 ::: ProductRelated_Duration (Gaussian)
-------------------------------------------------

ProductRelated_Duration    FALSE     TRUE
                   mean 1068.699 1881.240
                   sd   1650.812 2263.761
```

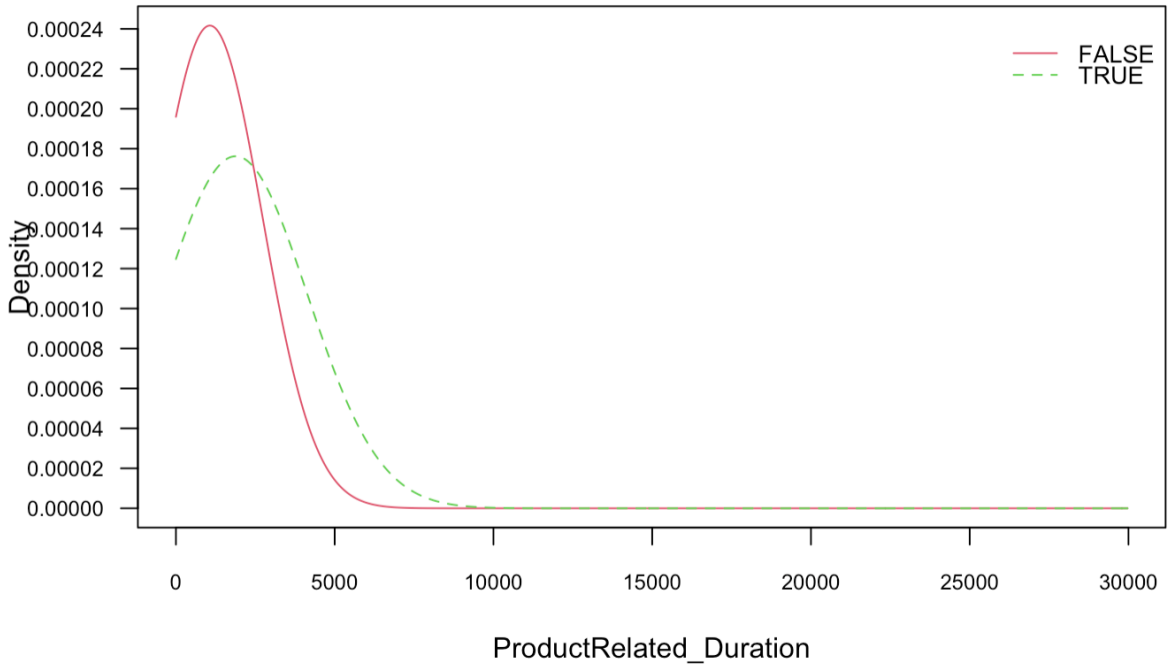

Figure 5.6: prior of feature "ProductRelated$_{D}uration$"

```
-------------------------------------------------
 ::: BounceRates (Gaussian)
-------------------------------------------------

BounceRates       FALSE        TRUE
       mean 0.025576282 0.005222718
       sd   0.052160920 0.012836500
```

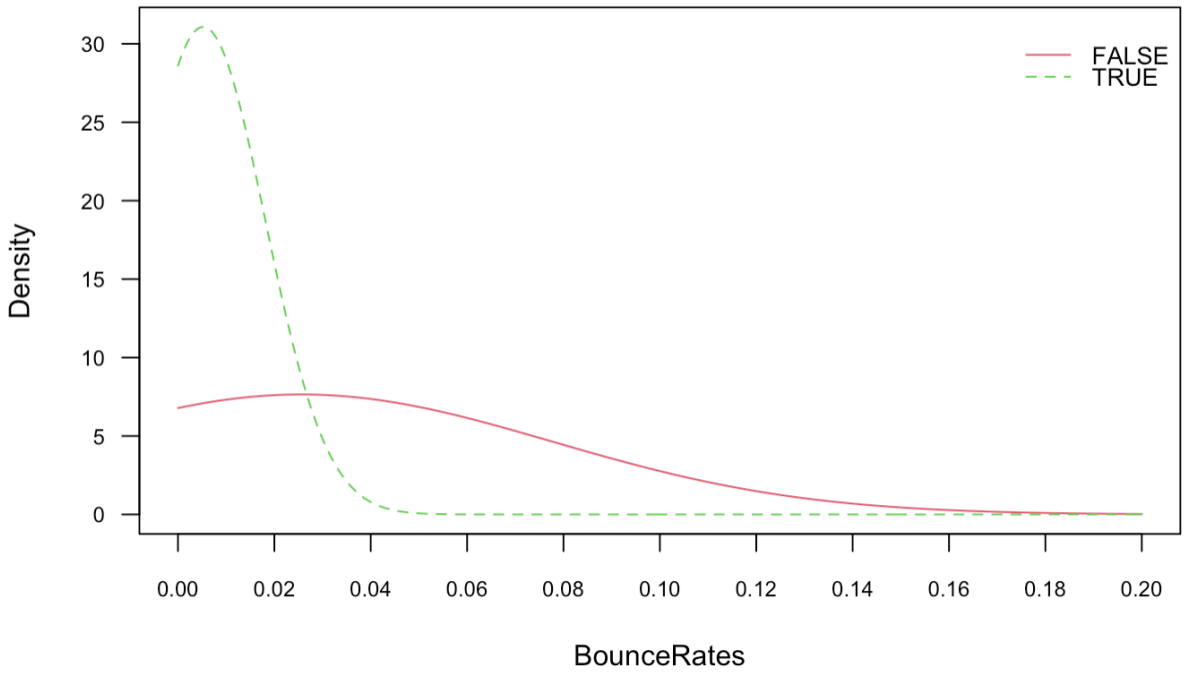

Figure 5.7: prior of feature "BounceRates"

```
------------------------------------------------
 ::: BounceRates (Gaussian)
------------------------------------------------

BounceRates       FALSE        TRUE
       mean 0.025576282 0.005222718
       sd   0.052160920 0.012836500
```

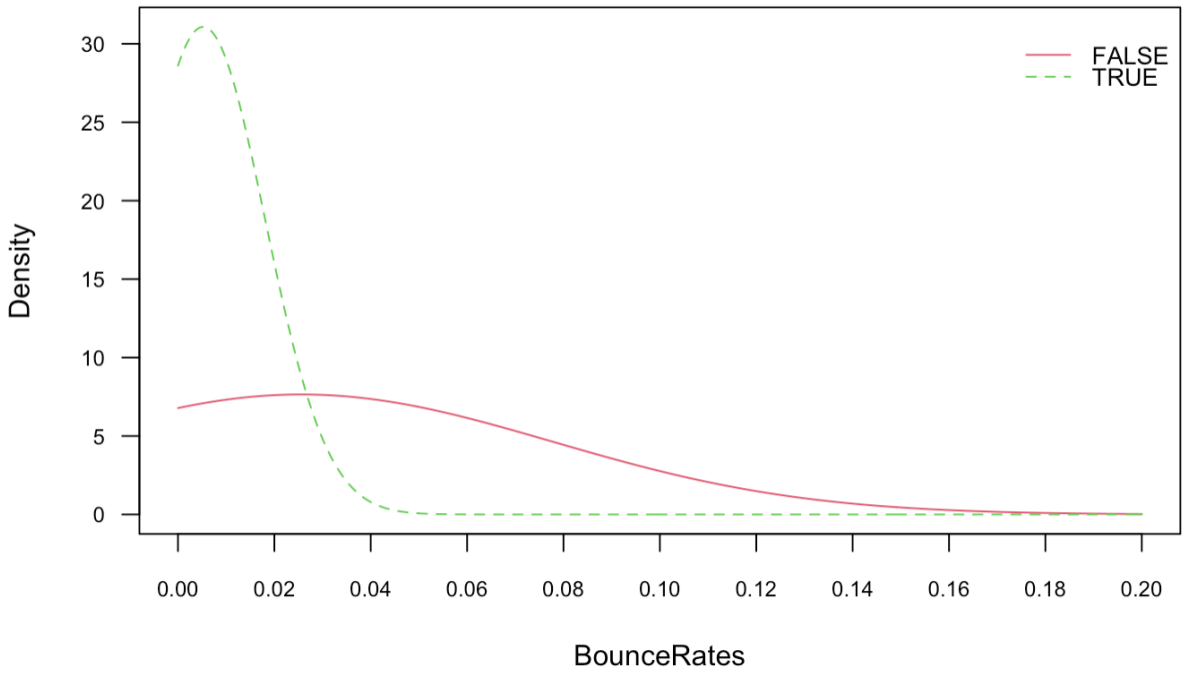

Figure 5.8: prior of feature "BounceRates"

```
------------------------------------------------
 ::: ExitRates (Gaussian)
------------------------------------------------

ExitRates      FALSE       TRUE
     mean 0.04760062 0.01966349
     sd   0.05141672 0.01689680
```

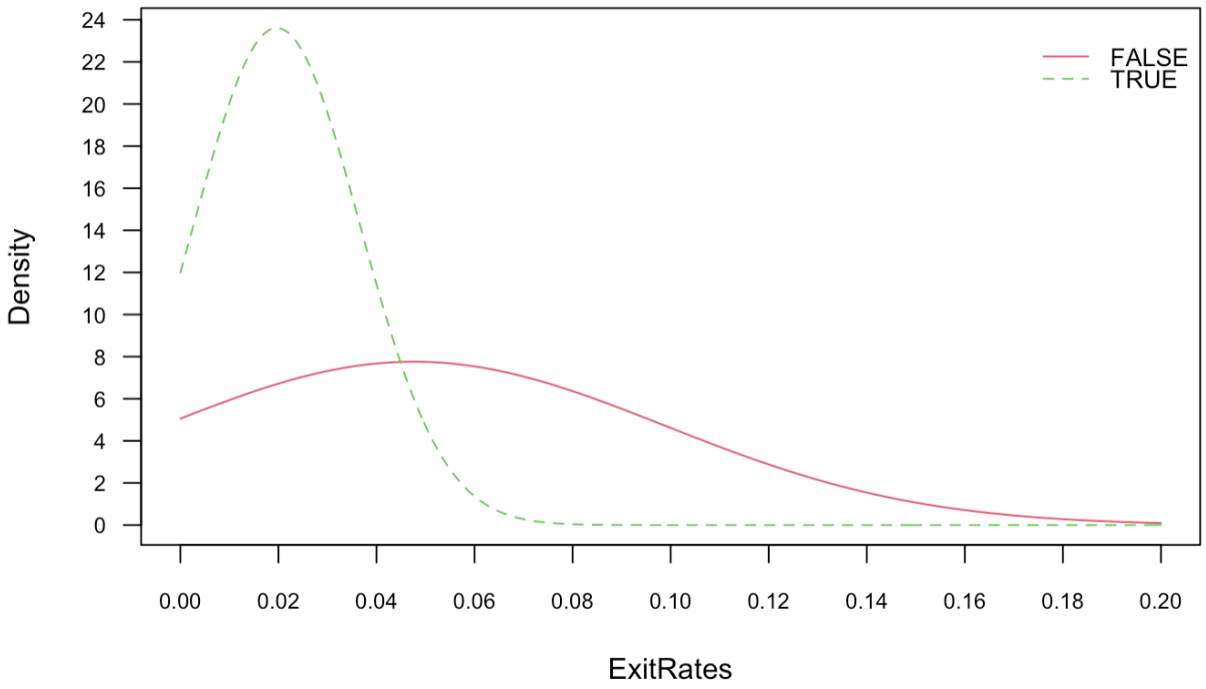

Figure 5.9: prior of feature "ExitRates"

```
-----------------------------------------------
 ::: PageValues (Gaussian)
-----------------------------------------------

PageValues      FALSE      TRUE
      mean   1.975970 27.470699
      sd     8.942083 35.711300
```

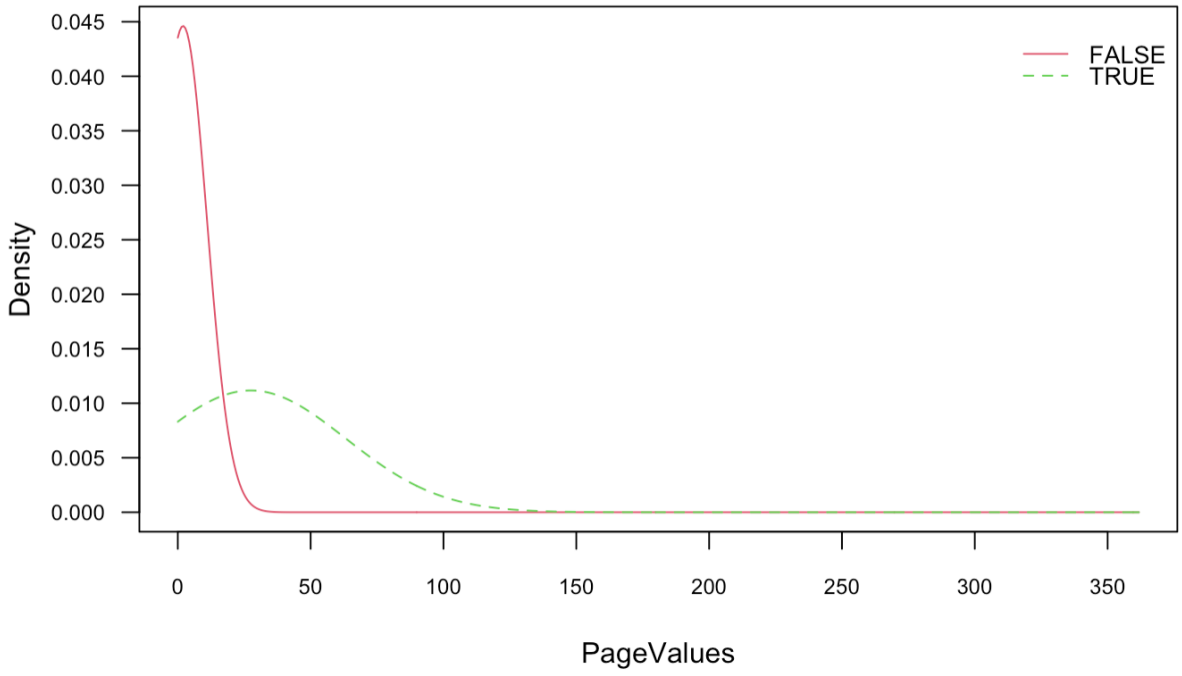

Figure 5.10: prior of feature "PageValues"

```
-----------------------------------------------
 ::: SpecialDay (Gaussian)
-----------------------------------------------

SpecialDay       FALSE       TRUE
      mean 0.06706644 0.02292076
      sd   0.20611279 0.12424813
```

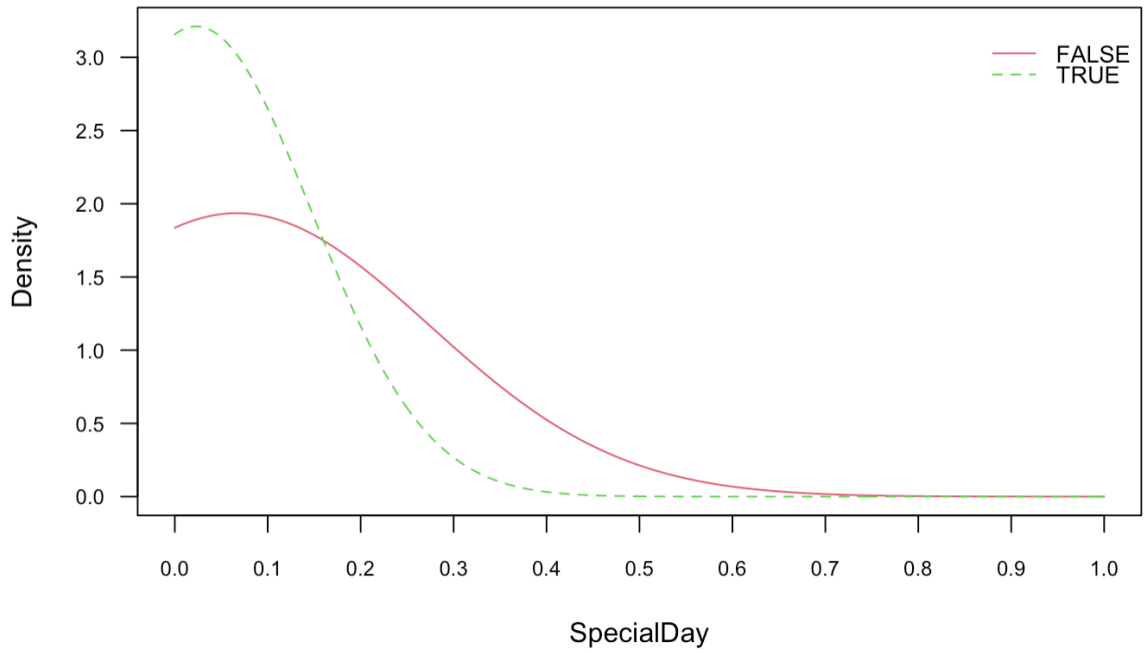

Figure 5.11: prior of feature "SpecialDay"

```
--------------------------------------------------
 ::: Month (Categorical)
--------------------------------------------------

Month        FALSE        TRUE
  Aug  0.035140321 0.037982973
  Dec  0.140681219 0.116568435
  Feb  0.017989926 0.001309758
  Jul  0.035500120 0.037328094
  June 0.023746702 0.014407335
  Mar  0.164907652 0.092337917
  May  0.287239146 0.194499018
  Nov  0.217798033 0.400785855
  Oct  0.042696090 0.064178127
  Sep  0.034300792 0.040602489
```

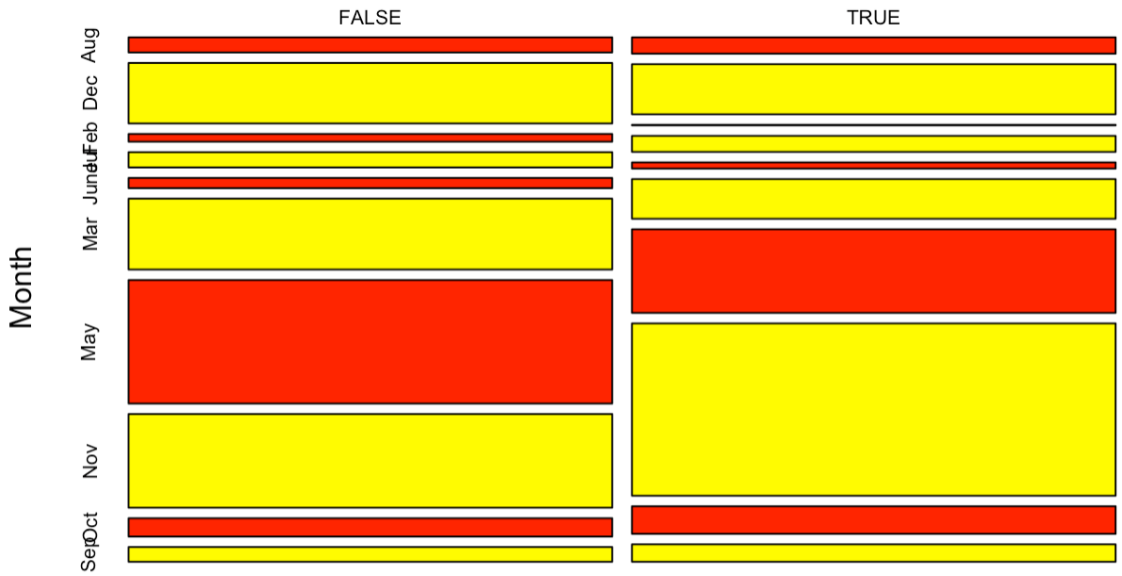

Figure 5.12: prior of feature "Month"

References

Czako Zoltan (2018) *SVM and Kernel SVM*, TowardsDataScience, https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200

Dan, S. A. (n.d.). *Support Vector Machines - II* [PowerPoint sildes]. UMB. https://www.cs.umb.edu/ dsim/cs724/ssvm2.pdf

Online shopping statistics you need to know in 2021. (2021, January 6). OptinMonster. https://optinmonster.com/online-shopping-statistics/

Understanding support vector Machine(SVM) algorithm from examples (along with code). (2020, December 23). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Vihar Kurama (2020) *Introduction to Naive Bayes: A Probability-Based Classification Algorithm*, Paperspace-Blog, https://blog.paperspace.com/introduction-to-naive-bayes/

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput Applic (2018). https://link.springer.com/article/10.1007/s00521-018-3523-0