E-Commerce Book Prediction Model

CS 577

Jared De La Serna

I. Abstract/Introduction

With the introduction of the internet, many aspects of our lives changed: how we obtain information, how we communicate, and the introduction of e-commerce. Before, commerce relied heavily on physical interactions such as physical stores and in-person transactions. With the introduction of the internet, e-commerce was introduced, changing how goods and services are bought and sold. Most of us have shopped online once in our lifetime. According to Forrester [1] , a leading global market research company, predicts that "Global Retail E-Commerce Sales Will Reach $6.8 Trillion By 2028," every year e-commerce sales increase around 30% every year. According to Statista Research Department [2], China leads in revenue in the e-commerce market with 1.3 trillion dollars, but after that comes China with 1.1 trillion dollars. As we can see, this is a market that has room for improvement. The fascinating fact about e-commerce is that anyone can sell a product online, opening doors to anyone interested in selling a product or a service. Since anyone can sell a product, e-commerce competition is high. Now, we must consider what makes a certain percentage of competition succeed. We must analyze patterns in e-commerce, and with those patterns, we can predict e-commerce results.

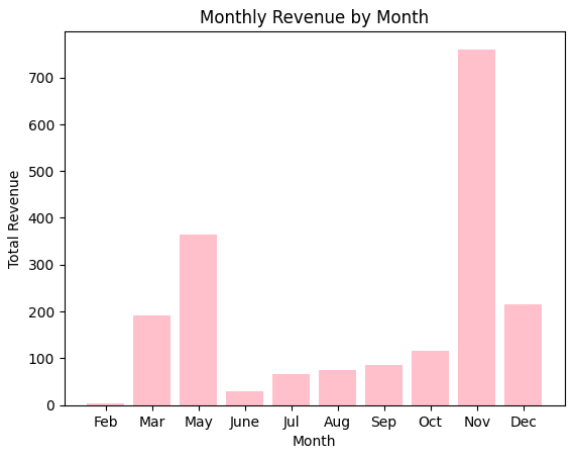
II. Approach

This dataset is from the UC Irvine Machine Learning Repository [3]. There's also clarification that "each session would belong to a different user in 1 year to avoid any tendency to a specific campaign, special day, user profile, or period". The dataset was obtained from the OSCommerce platform. The dataset is from an online bookstore that has 12,330 sessions. The dataset is about book purchases, which is essential to understanding the context better. The main goal of this project is to see if we can analyze the patterns and use them to create a model to predict the rate of shopping revenue with the highest success. To do this, we must do the following task before we start with the dataset: we will have to make sure that we clean the dataset, and then we will do Exploratory Data analysis to identify patterns, distributions, and relationships that we can patterns. We can use probabilities to

get a mathematical sense or correlation between variables, especially the probabilities of purchases based on certain factors. We will use Pandas to properly load our data, calculate total and average Page Values per region, and find the top traffic sources. We are contributing purchases. We will also use Gradient Descent to optimize weights for logistic regression and visualize the convergence of the cost function. We must use the method “One-Hot Encoding” to successfully apply the gradient descent. We will also use linear regression to predict Page Values based on other features. We will use the ROC Curve to evaluate this model. We will also do a logistic regression because certain of our features are categorical variables. We can also create and train a Decision Tree classifier to predict revenue, which we can visualize decision trees to get a better understanding of the features important to get revenue. Finally, we will do cross-validation to compare model performance.

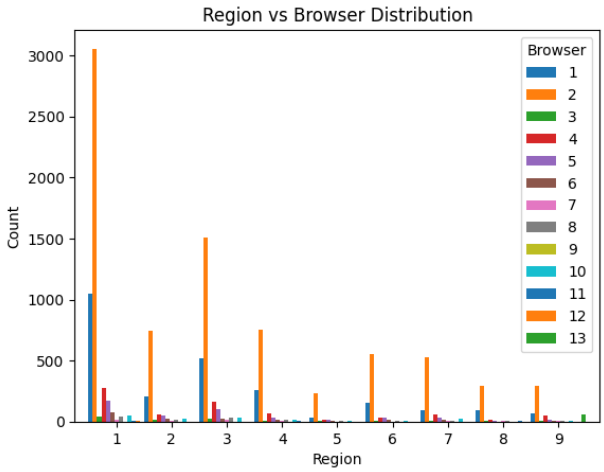
III. Data Analysis

In this study, we used many tools that Python provides. The first tool we used was a built-in function from Python to count all our missing values (NaN). We used the is. null() and .sum() functions. This showed us how many missing values our dataset had; surprisingly, our dataset had no missing values. For the explanatory data analysis, we used Matplotlib; Matplotlib is used mainly to create visualization, customize plots, and analyze data visually. Is there a correlation between revenue success and the month of the year? To clarify, this dataset has no values for January and April. The first thing we did was group the data by the month columns to calculate the total counts of revenue. We used functions grouby() and sum(). We had to put the months into the audience to understand it more easily. We accomplished this by using re. index, and declaring our specific order. Then we did plt.bar(), and then we used Matplotlib to add a label to the title, xlabel, ylabel.



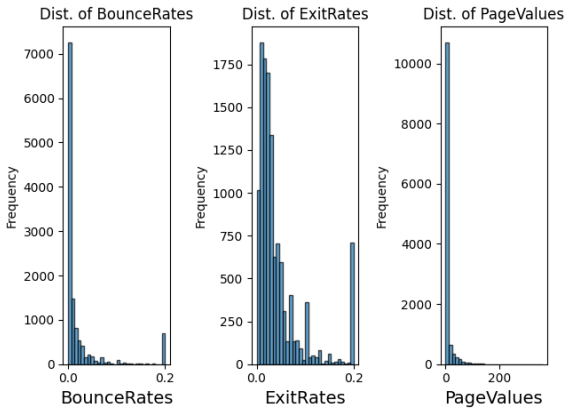
There is a peak in November. This trend can be explained by the "Black Friday" tradition and "Cyber Monday." At first, a peak in March and May seems reasonable because

there are no major holidays, but after researching, there's a reason. March is considered Nation Reading Month, which encourages reading activities. The second Sunday in May is Mother's Day, which supports the idea of people gifting their mom's books. According to the graph, the best months for selling books are March, May, November, and December. These months had the highest revenue sessions. Since advertising is important, we must know what regions and types of browsers they use. This could be important and could help us set up ads more significantly. The geographic region of the buyer is important because it could mean that specific areas buy more books than others. I wanted to see what browser they use the most; this is useful to set up ads. I did the relationship contingency table between two categorical variables, region, and browser, using a bar plot. I used matplotlib to create a contingency table. This was done by using the contingency_table() function.



As we can see, Browser 2 was the most used in all regions. As we can see, Region 1 was the most popular browser, and also, we can see that in Region 1-3, there are a lot of online book shoppers. Now, we have features that describe the online shopper's behavior, such as bounce rates, exit rates, and page values. The bounce rate is the proportion of visitors who open the website and leave without interacting with the website. Exit rates are the percentage of visitors that leave the book website from a specific page after viewing multiple books. Page Values is the website's value; for example, 0 means that the customer just opened it and correlated with any purchases; the higher the value, the better. I wanted to see the distribution of each variable and how the customer shops. I stored those variables in an array, created a loop to run through those three variables, and created a histogram with 30 bins. I used plt. Hist ()

will create a histogram distribution showing us the frequency.

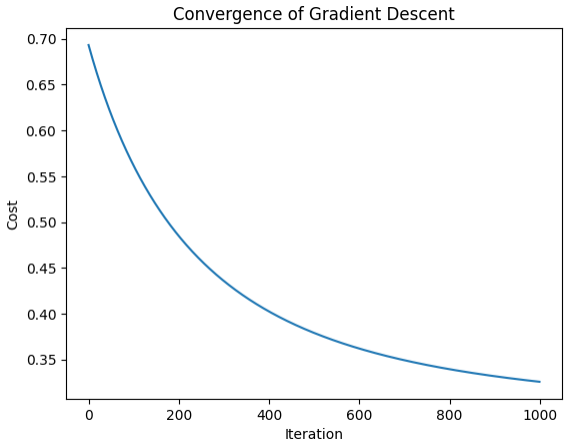


As we can see, most of the BounceRates values are near 0, implying that most of the shoppers just leave the website without interacting with any other pages. The highest value of BounceRates is that most shoppers who buy have similar bounce rates. With the ExitRates Distribution, most of the values are closer to 0, which implies that most book shoppers leave the website after just being interested in a little bit. Most of the sessions end with an ExitRates end at 0.2, similar to BounceRates. With the PageValues distribution, we can see that most of the page values are at 0, implying that people don't buy when they just open the first site, go through most pages, and then buy.

IV. Gradient Descent

To do gradient descent, we transformed a non-numeric into a numeric column. First, we used select_dtypes(include= Object') to store non-numeric columns. Since Weekend and Revenue columns, are TRUE or FALSE values, they were converted into 0 or 1 values using the. astype(int) function. The rest of the remaining non-numeric columns were Month and VistorType. For these numeric columns, we did **One-Hot Encoding**. Using the pd_get_dummes() function to One-Hot Encoding after this, our dataset had all numeric values. Now, our dataset is ready for gradient descent. Gradient descent doesn't accept categorical variables, so one-hot Encoding is necessary. We decided to do gradient descent. We used the sci-kit-learn library because it has a collection of modules that we can use for the gradient descent. It has modules such as train_Test_split used to train and split the data, **StandardScaler** to preprocess the data, accuracy_score, precision_score, and recall_Score are used to evaluate the model. In the logistic regression approach using gradient descent, we used Feature-Target Separation. Revenue is our Target Variable (y); the rest are input features (X). We **standardize** the features for a better

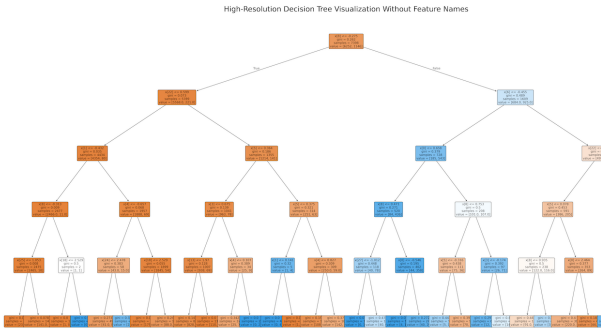
gradient descent performance and **standardize** the features, implying that the mean is equal to 0 and the standard deviation is equal to 1. We split the data into 60% training and 40% testing sets. We defined a sigmoid function. We created a gradient descent function with cost tracking. This code was mostly obtained from the lecture notes on gradient descent logistic regression. We initialize weights and biases to 0. The cost function was the binary cross-entropy loss, the difference between predicted probabilities and true labels. Since our target Revenue is 0 or 1. We created a prediction function that classifies it and uses 0.5 as a threshold. We used the functions accuracy_score(), precision_score, recall_score() and our paraments were y_Test and y_test_pred. Our **results** were 87.8% accuracy, 76.2% precision, and 30.2% recall. The model correctly predicts whether a customer will purchase, at 87.8%. The model predicts that the customer will make a purchase 76.2% correctly. The model identifies only 30.2% of actual buyers correctly. The recall is pretty low, but we can justify this due to the fact that most of the customers are more likely not to purchase a book. To see if we did the gradient descent correctly, we need to see the cost converge through iterations.



We can see the curve converges, meaning we have correctly done logistic regression with gradient descent.

V. Decision Tree

To further our models evaluation, we can do DecisionTreeClassifier. A decision tree classifier is a machine learning algorithm used for classification tasks. We split the dataset into branches based on feature conditions to predict target variables. Since we have categorical variables and conditions that make sense in online shopping, it can show us that there are patterns and rules to understand what drives book purchases, giving us more information about the user's website behavior. Here's our tree



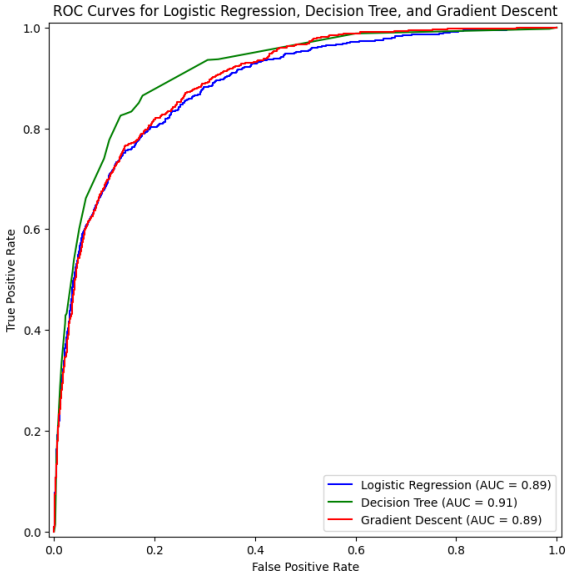
The decision tree predicts customer purchase behavior on features like PageValues, which indicates the value of the pages visited, and BounecesRates, which reflects single-page decisions. These splits can help us identify high-value customers versus those who don't make a purchase.

VI. Logistic Regression

Now we can use sklearn. Linear model to logistic regression, a similar process as we did in the gradient descent, but we import the package LogisticRegression to use the function LogisticRegression(). Our results are 88.2% accurate, whatever a customer will purchase that specific session. Our precision was 75% predicts that a customer will make that purchase. Our recall was 34.9%, identifying only 34.9% of the actual buyers, but a similar reason to the gradient descent low recall can also justify this.

VII. Evaluations

To compare our models, we can use cross-validation, in which a subset can be the ROC Curve because the ROC Curve is a visual of the model performance. The 3 models, logistic regression with gradient descent, logistic regression, and decisions:



We can safely conclude that our decision tree has the best performance in distinguishing between customers who will buy a book and those who won't. We can safely conclude that if we want to have the highest prediction model is that we should focus on how PageValues works because it is the most influential, and we can say that having interactive initiatives to shop when the shopper opens the book website is crucial. For example, SpecialDays also has a good influence; users are more likely to buy books during holidays. BounceRates and ExitRates are also crucial in forming a correlation with PageValues. TrafficType

predicts purchases by identifying high-intent users based on their source (e.g., organic or referral). At the same time, VisitorType separates likely buyers, with returning visitors being more likely to purchase than new or casual visitors. [4]

VIII. Unique Project

The founders of this dataset used this project to write a research paper on machine learning algorithms like Randon Forest, Support Vector machines, and Multilayer Perceptron to predict purchasing intent. What I did differently was I decided to use Logistic Regression with and without regression, and I did Classification with a Decision Tree, my work uses packages and libraries that are useful for machine learning, while the founder of this dataset t focused more on heavy statistic computing,

IX. Motivation Statement

My motivation for this project is to improve my knowledge of phyton libraries. I'm interested in online shopping to an extent, but I believe that you work on a project when you lack knowledge it can provide skills you haven't learned yet. For example, I'm used to doing mostly numeric prediction models instead of categorical. After this project, I can see how to work with classification problems. For example, I couldn't figure out why there was a trend in shopping books in March and May; it made me think outside the box.

REFERENCE:

[1] “Global: E-Commerce Revenue by Country 2022.” *Statista*, www.statista.com/forecasts/1283912/global-revenue-of-the-e-commerce-market-country.

[2] Miglani, Jitender. “Global Retail E-Commerce Sales Will Reach $6.8 Trillion by 2028.” *Forrester*, 21 May 2024, www.forrester.com/blogs/global-retail-e-commerce-sales-will-reach-6-8-trillion-by-2028/.

[3] Sakar, C. & Kastro, Y. (2018). Online Shoppers Purchasing Intention Dataset [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5F88Q.

[4] “The 6 Best Ecommerce Advertising Strategies for 2022.” *Omnisend Blog*, 30 Apr. 2021, www.omnisend.com/blog/ecommerce-advertising/.