Data Analysis of Online Shopper's Purchasing Intention

MSBA-326: Machine Learning for Predictive Analytics

Professor: Bahman Zohuri

Group Members:

Andrew Frazier, Fatbardha Maloku, Xinzi Li, Yichun Chen, Yeji Jung

TABLE OF CONTENTS

ABSTRACT

In an era of widespread internet-based commerce, any company with a web-based storefront is looking for ways to improve the customer experience, with the ultimate goal of facilitating purchases. This process can take many forms and use many strategies, but all of them start with one core task. The company must be able to identify who is least likely and most likely to make a purchase. We suggest that with basic web browsing analytics, machine learning can provide this capability, and is a viable tool for effective customer segmentation.

INTRODUCTION

Retail shopping is continuing to shift to the E-commerce format and as a result, the dynamics of shopping, and how companies facilitate purchases, are changing. E-commerce has become the dominant form of the retail marketplace. Online customers often browse a variety of pages on a particular E-commerce site before placing an order or making a purchase. However, most of the time, customers who visit these online sites may not make any purchases at all. This could be for distinct reasons, product pricing or window shopping. It is essential to predict a customer's purchasing intention so that retention measures, such as recommending the right products, can be taken to convert the potential customer into a buyer. This information can help businesses better cater to customer preferences and increase sales through various marketing techniques depending on the customer's likelihood of a purchase. Here the prediction of customer purchase intention can help strategize different marketing strategies and could be added to the mechanism of the recommendation system of an e-commerce retailer. (Tai, 2013)

PROBLEM STATEMENT

Problem Statement

The main problem that most e-commerce companies can face is that the customer does not purchase the item or complete their payment after visiting the website, clicking, and browsing the items for a long time. The reasons can vary. Olenski (2017) stated that the average conversion rate of the website is only 2.35%, and it is very crucial to find out possible ways to turn visitors into buyers. The following is a list of some reasons why the business can be below potential:

- Visitors are not asked to buy
- The buying process is complex
- No sufficient customer support to answer shoppers' questions
- Customer does not find what they need
- The website is not secure enough
- Targeting the wrong people

Correctly targeting the groups of customers likely to make a purchase is the driving business context behind our project. We intend to create an ML model that can identify these users based on historical browsing and purchase data, which would theoretically allow us to pass this information on to other departments who can then target them with incentives and other special offers in order to further increase the likelihood of a purchase.

Problem Background

According to Couto et al. (2022), machine learning already in use has proved to have a strong impact on consumer segmentation particularly for the company that relies primarily on internet-based storefronts. As an example, we have eBay, Amazon, and Alibaba, the biggest company around the world. Alibaba has focused exclusively on e-commerce and has implemented AI-driven data from their entire sales system. Even though machine learning and AI can provide a powerful solution for an e-commerce company's challenges, if the company does not know how to use them effectively, it cannot extract actionable insights from the results. Using predictive technologies and analytical tools can be one of the methods that e-commerce-based companies use to find patterns and trends in consumer browsing behavior. Couto et al. (2022) pointed out that many retail and e-commerce companies use recommendation engines to boost sales and customer satisfaction by providing personal recommendations and convenience which can lead to higher sales with conversion rate and retention of customers.

Project Goal

Our team will use a dataset containing consumer browsing and purchase activity from an anonymous website to predict online shoppers' purchasing intentions. We will evaluate patterns in the dataset by using descriptive statistics and exploratory data analysis and then use the insights we find to create a machine learning model that can predict whether a consumer will make a purchase or exit the webpage without making a purchase.

LITERATURE REVIEW

Online shopping is anticipated to be used by 2.14 billion individuals worldwide in 2021. (Coppola, 2021). At the same time, global e-commerce sales are forecast at $4.891 trillion. If these online shopping statistics aren't enough to blow you away, projections show that e-commerce sales worldwide are going to grow to $6.4 trillion by 2024. The biggest online marketplace is the Chinese platform Taobao, with a GMV of $515 billion. Companies such as Tmall and Amazon landed in second and third place, with third-party GMV of $432 billion and

$344 billion, respectively. To put this figure into context, the top online marketplaces worldwide sold $1.66 trillion worth of goods in 2018. More than 50% of all online purchases were made on marketplace websites like those run by Alibaba, Amazon, and eBay in 2018 (Mohsin, 2021).

The impact of the coronavirus has probably contributed to one of the major changes in consumer behavior this year. The number of people shopping online, especially for groceries, has surged as a result of the severe lockdown measures that are being implemented by nations throughout the world in an effort to stop the virus' spread. A recent study found that 42% of Americans bought groceries online at least once a week in March 2020. (Soper, 2020). This is a huge improvement from the 22% seen just two years ago. Daily grocery sales on the internet have also climbed by as much as twofold.

APPROACH METHODOLOGY

We follow a standard data analytics workflow for this project. After data ingestion, we use descriptive statistics to identify columns and key metadata to identify any areas that require cleaning. We then utilize Exploratory Data Analysis to determine any immediately obvious patterns or trends. Finally, we use the insights gathered from the first 2 steps to build an ML model that can predict whether a website visitor will make a purchase.

Machine Learning Libraries and Tools

The machine learning tools used in this project rely almost exclusively on Scikit-Learn in addition to the standard data analytics libraries found within the python ecosystem (Pandas, Numpy, etc). Scikit-Learn provides an extensive array of classification models that we can test for optimal performance in our use case, as well as tools that allow us to tune parameters based on a specific scoring function and interoperability with several other libraries. The other notable library used for the purposes of this use case is the Imbalanced-Learn library, which augments

Scikit-learn's ability to handle highly imbalanced target classes, which we have in the selected dataset. Further details regarding the machine learning process are explained in future sections.

SOLUTION PROCESS

Dataset Metadata and Descriptive Analysis

We use a dataset provided by the UCI Machine Learning Repository that contains data related to web browsing activity on an anonymous e-commerce site. It provides us with multiple features that describe types of pages visited, visit duration, and demographics surrounding geo, browser, OS, etc. It contains 12,330 observations, across 18 features.

Data Dictionary

<u>Numeric Features</u>

- `administrative` - number of admin pages visited
- `informational` - number of informational pages visited
- `productrelated` - number of product-related pages visited
- `administrative_duration` - duration of visit on page type in seconds
- `informational_duration` - duration of visit on page type in seconds
- `productrelated_duration` - duration of visit on page type in seconds
- `bouncerates` - the percentage of visitors who enter the site from that page and then leave without visiting any other pages.
- `exitrates` - the percentage of visits to a webpage that was the last in that session, compared to all visits to that webpage.
- `pagevalues` - the average value for a web page that a user visited before completing an e-commerce transaction

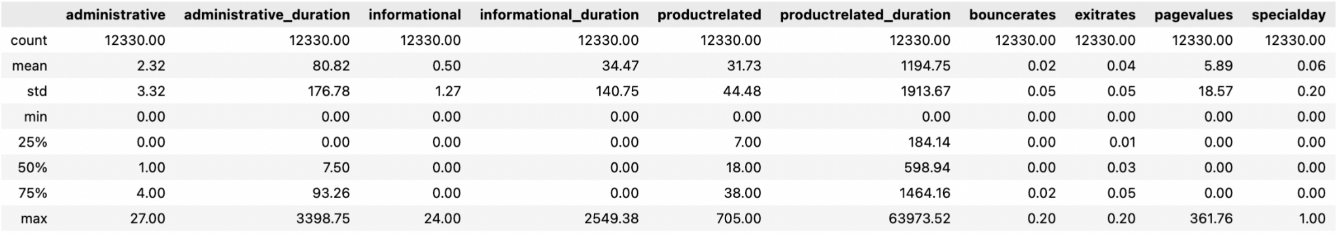
- `specialday` - numeric representation of proximity to holiday

Categorical Features

- `month`
- `operatingsystems`
- `browser`
- `region`
- `traffictype` - where the visit originated from external to the site itself.
- `visitortype` - new/returning visitors
- `weekend`
- `revenue` (Boolean Target Feature)

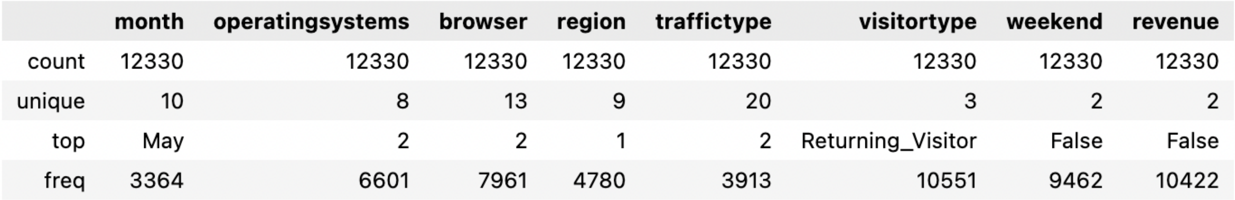
For our analysis, we consider how each of the variables impacts the outcome of a given consumer's shopping behavior. The powerful statistical toolsets available in python allow us to accomplish this more easily. In the example below, we've used the describe function to learn more about the count, mean, standard deviation, minimum, and maximum values of our data set as well as the summary statistical analysis of the columns in the online shopping data collection. Our target variable in this analysis is the column "Revenue". We will further identify how the explanatory variables in the dataset influence and correlate with the target variable.

Numeric Features:



	administrative	administrative_duration	informational	informational_duration	productrelated	productrelated_duration	bouncerates	exitrates	pagevalues	specialday
count	12330.00	12330.00	12330.00	12330.00	12330.00	12330.00	12330.00	12330.00	12330.00	12330.00
mean	2.32	80.82	0.50	34.47	31.73	1194.75	0.02	0.04	5.89	0.06
std	3.32	176.78	1.27	140.75	44.48	1913.67	0.05	0.05	18.57	0.20
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00	7.00	184.14	0.00	0.01	0.00	0.00
50%	1.00	7.50	0.00	0.00	18.00	598.94	0.00	0.03	0.00	0.00
75%	4.00	93.26	0.00	0.00	38.00	1464.16	0.02	0.05	0.00	0.00
max	27.00	3398.75	24.00	2549.38	705.00	63973.52	0.20	0.20	361.76	1.00

Categorical Features:



	month	operatingsystems	browser	region	traffictype	visitortype	weekend	revenue
count	12330	12330	12330	12330	12330	12330	12330	12330
unique	10	8	13	9	20	3	2	2
top	May	2	2	1	2	Returning_Visitor	False	False
freq	3364	6601	7961	4780	3913	10551	9462	10422

Exploratory Data Analysis (EDA)

We employed the visualization methodology to delve deeper into the variables and their distribution throughout the EDA process. We have gained valuable insights from the visualizations that will greatly aid our subsequent predictive analysis.

Target Label Distribution

The target variable "Revenue" contains boolean values, where "True" means a purchase was made for revenue generation, while "False" means no purchase was made and thus no revenue was generated. It suggests that our machine learning model needs to solve a binary classification problem. Figure 1 below shows an imbalanced distribution of the target label. The target indicates that out of all consumer browsing sessions, 85% resulted in no purchase, and 15% resulted in a purchase. We'll learn more about this class imbalance, and the mitigation steps for it in the modeling portion of this paper.

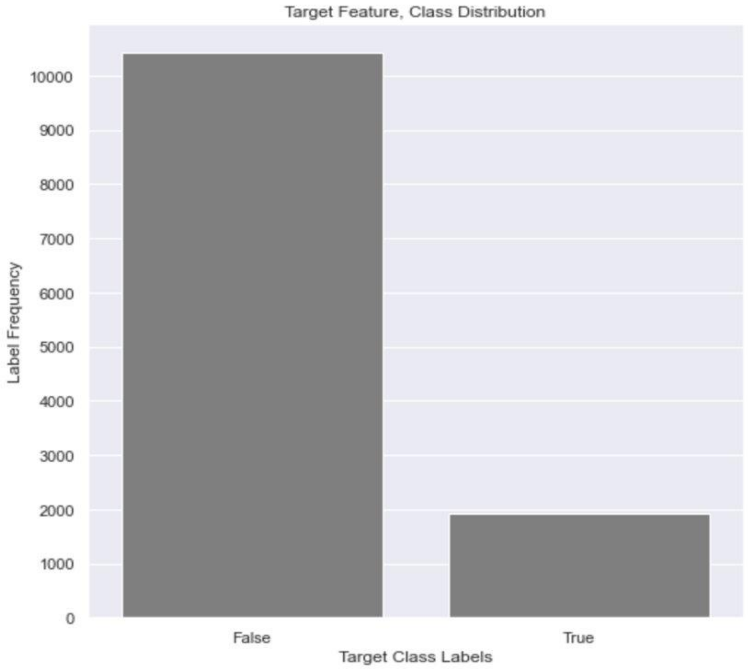


Figure 1: Target Label Distribution

Categorical Features EDA

We explored the impact of each of the eight (8) categorical features on the class label, mainly from two different angles - “frequency” and “probability”. Due to page limits, we only include discussions of a few of them in this paper as follows.

Month. Figure 2 below shows the impact of the month on revenue generation. The left “frequency” bar chart showed that some months (March, May, November, and December) have greater activity than other months, but it's difficult to tell if any month contains a greater probability of a purchase or not. In order to obtain this insight, a “probability” chart was created (see the right portion of Figure 2). We can see much more clearly that while in general, a month seems to have very little impact on revenue generation (the highest probability is only just over 20%), some months (October and November) do have an increased probability of purchases as compared to other months.

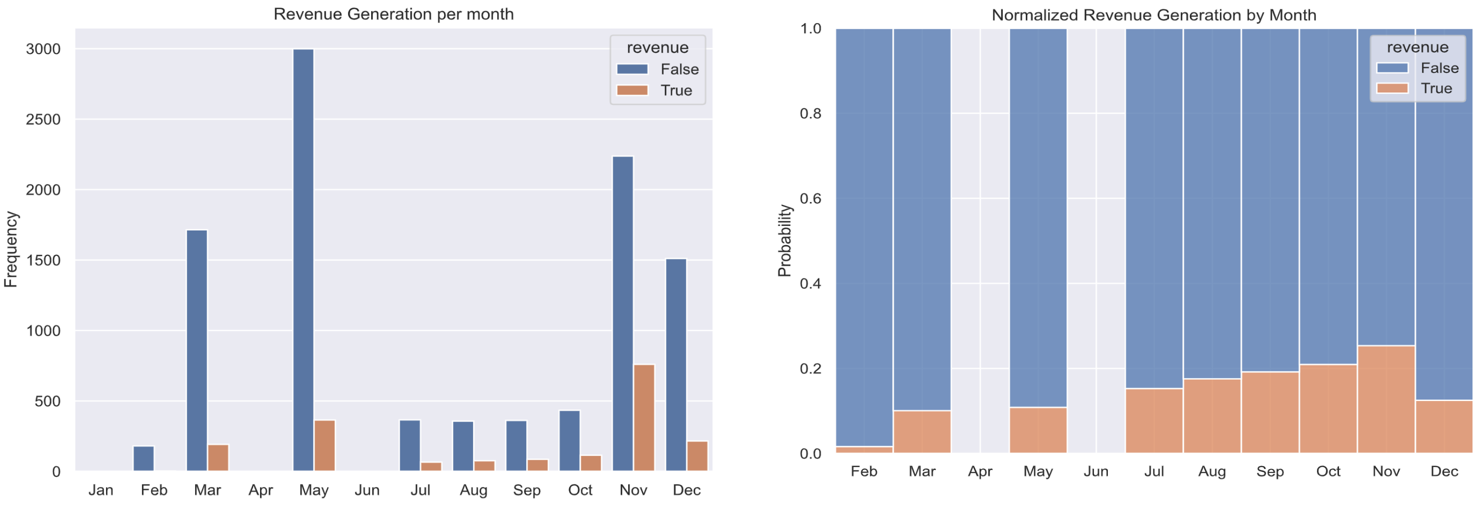


Figure 2: Revenue Generation Vs. Month

Operating System (OS). Figure 3 below shows the impact of the operating system on revenue generation. The left “frequency” bar chart showed that customers are largely using a limited selection of OS, but similar to the 'month' bar chart above it’s difficult to know whether any given OS has an impact on the likelihood of a purchase. A normalized view on the right shows that the OS also has seemingly little impact on revenue generation.

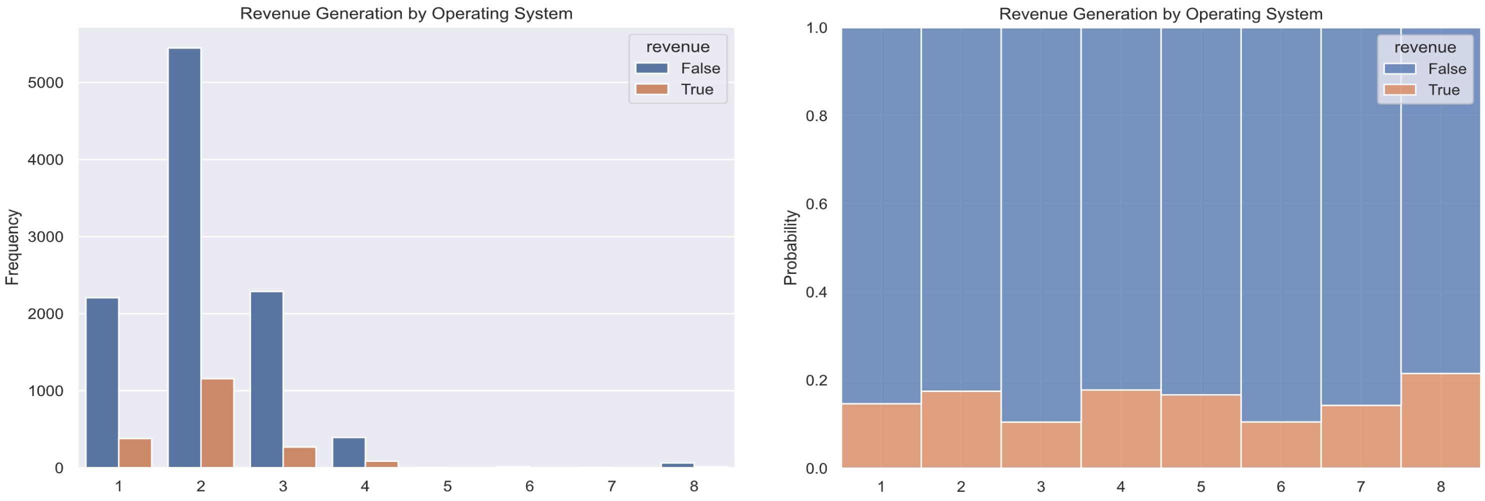


Figure 3: Revenue Generation Vs. Operating System

Visitor Type. Next, we looked into the "visitor type" variable. The bar chart that follows displays the revenue generation for three different visitor types. About 13.9% of visitors are new, while 85.5% are returning customers. There is also a small percentage of about 0.7% of other types of visitors that are shown in the below visualization. We can see that returning visitors generate a higher volume of website activity and have more purchases than the other two visitors

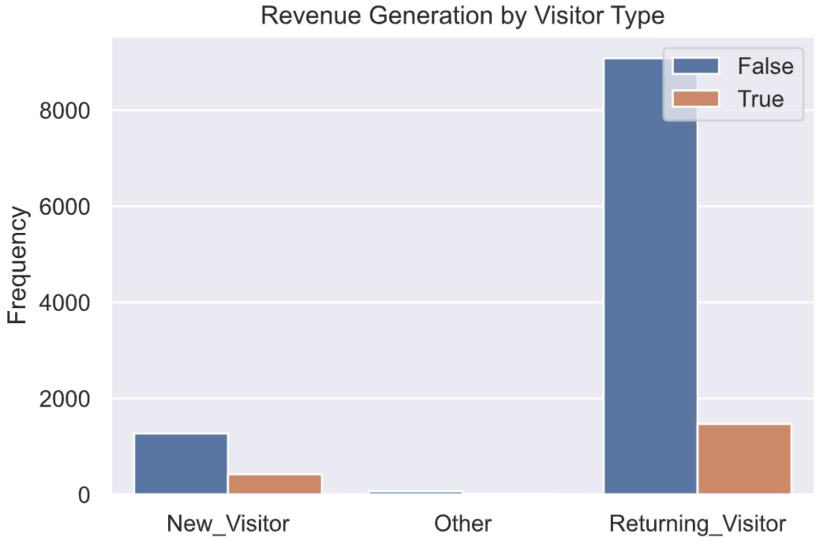
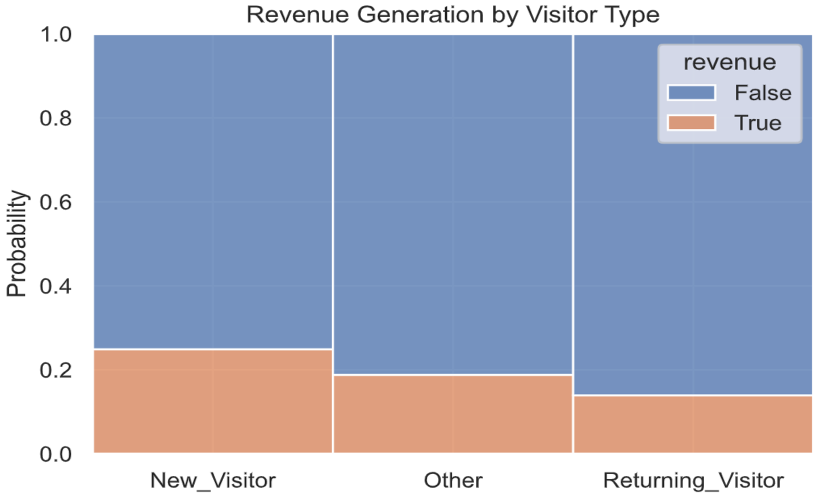
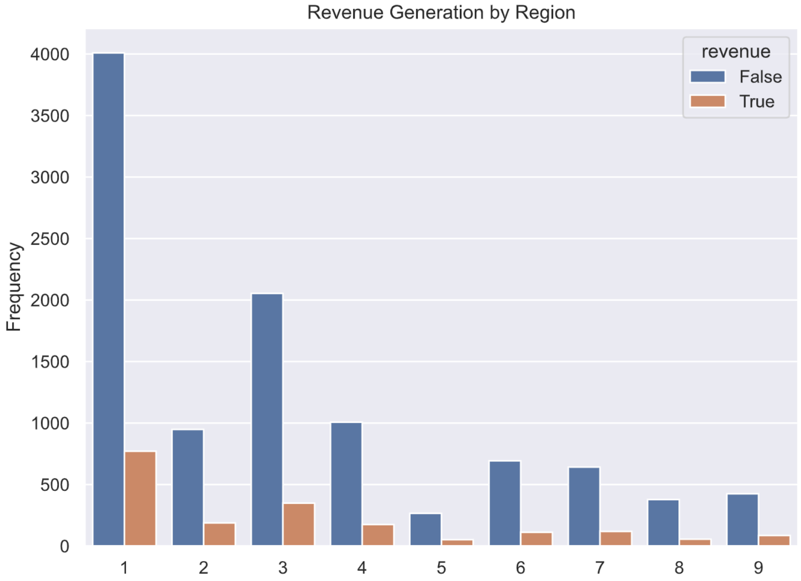


Figure 4: Revenue Generation by Visitor Type

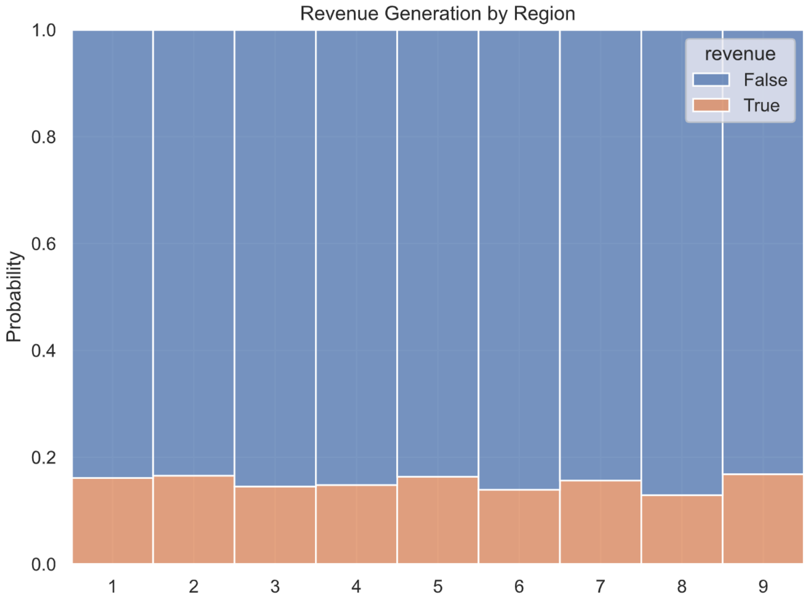
From the "probability" angle, however, we found that new visitors have a higher probability of purchasing an item than the other two visitor types, but this is by a narrow margin.



Region. The analysis of the region variable also provides us with valuable insights. The figure below displays various geographical areas from which websites were accessed. It can be seen that most website visits come from Regions 1, 2, 3, and 4, while only a small number come from Regions 5, 6, 7, 8, and 9.



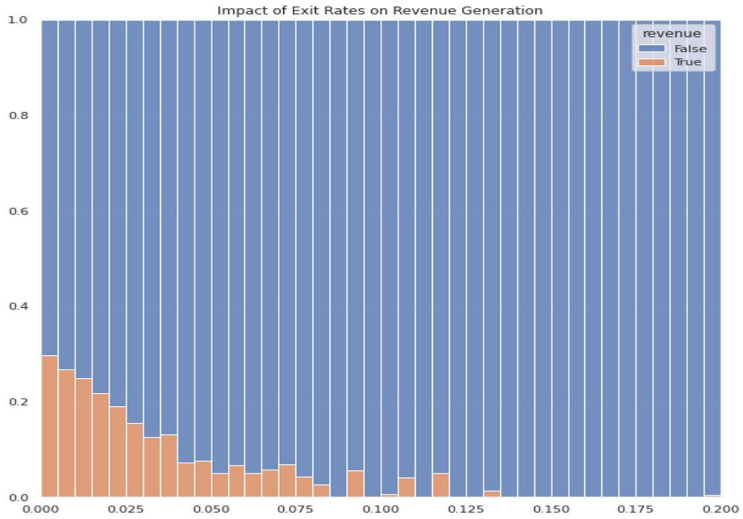
However, the following “probability” chart suggests that the probability of revenue generation is not very different among all the regions.



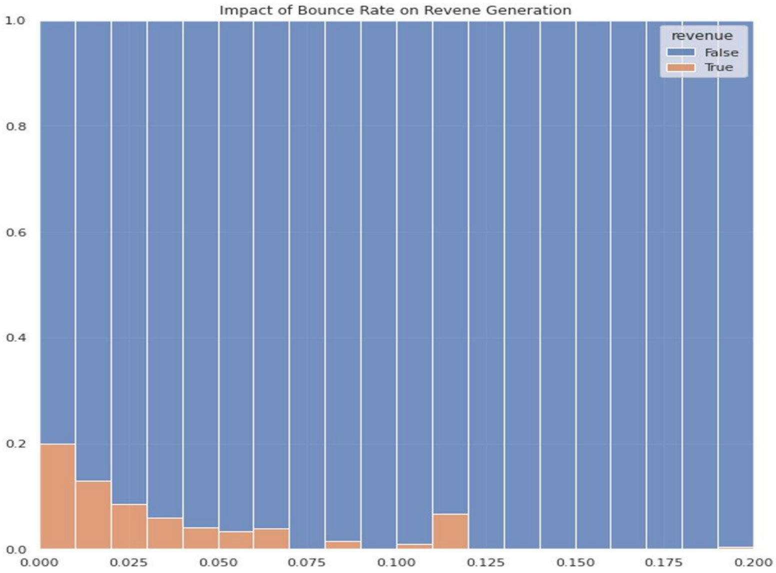
Numerical Features EDA

We explored the impact of each of the ten (10) numerical features on the target class using a combination of “frequency”, “probability” histograms, and sometimes scatterplots. Due to page limits, we only include discussions of a few of them in this paper.

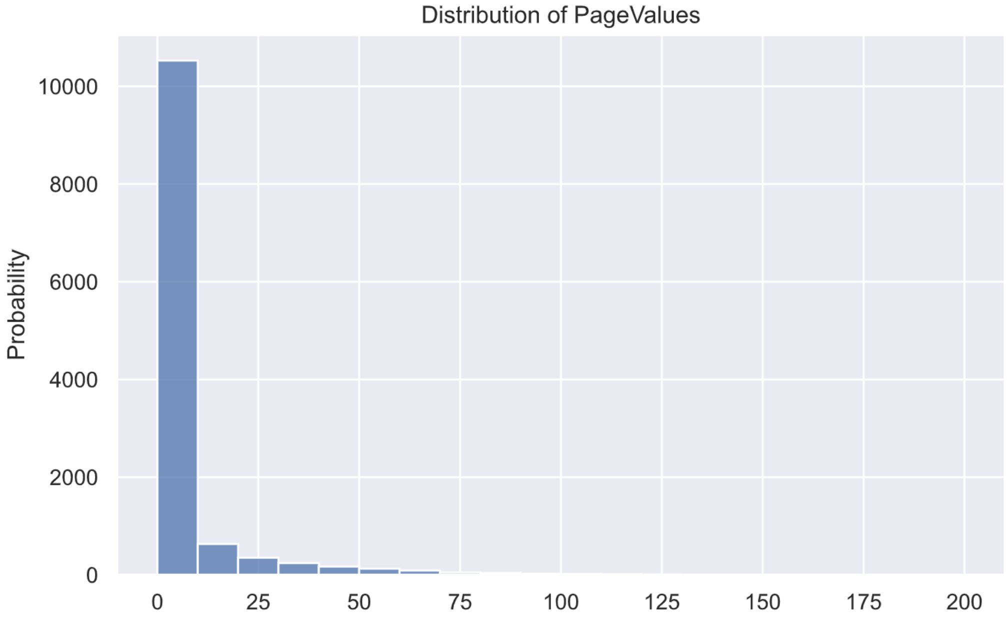
Exit Rate. The impact of exit rate on revenue creation is seen in the graph below. A lower exit rate is strongly correlated with a higher likelihood of generating income.

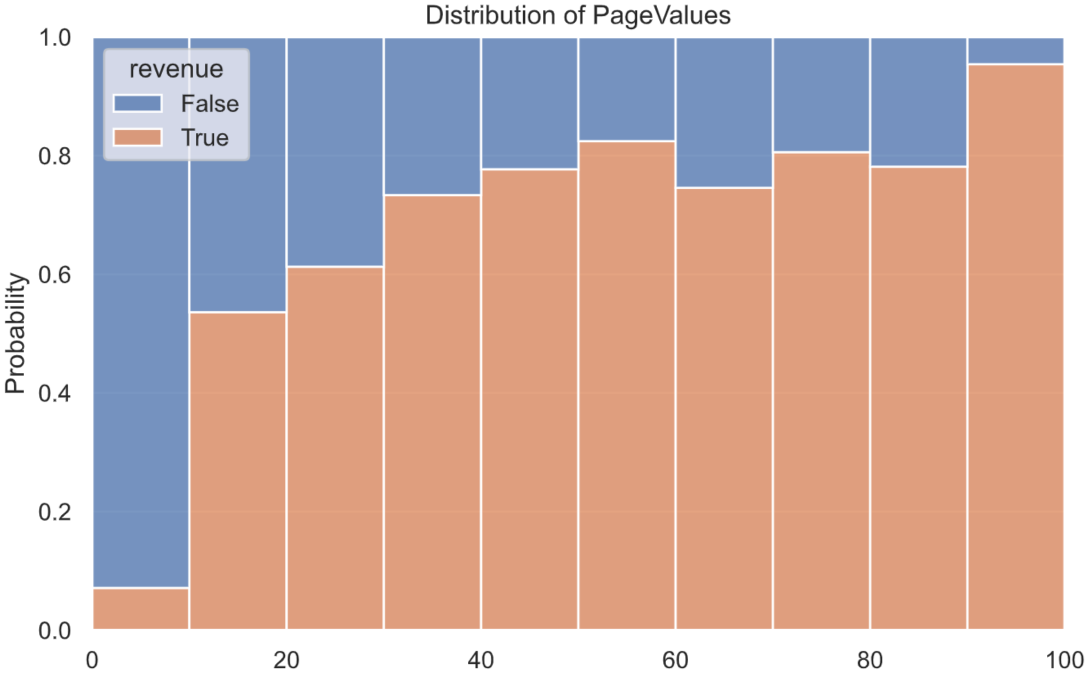


Bounce Rate. As expected, bounce rate has an inverse correlation with revenue creation.



Page Values. We can clearly see that the distribution is essentially limited from 0-100, so we'll limit our probability view to this range to avoid skewed data from a lack of samples.

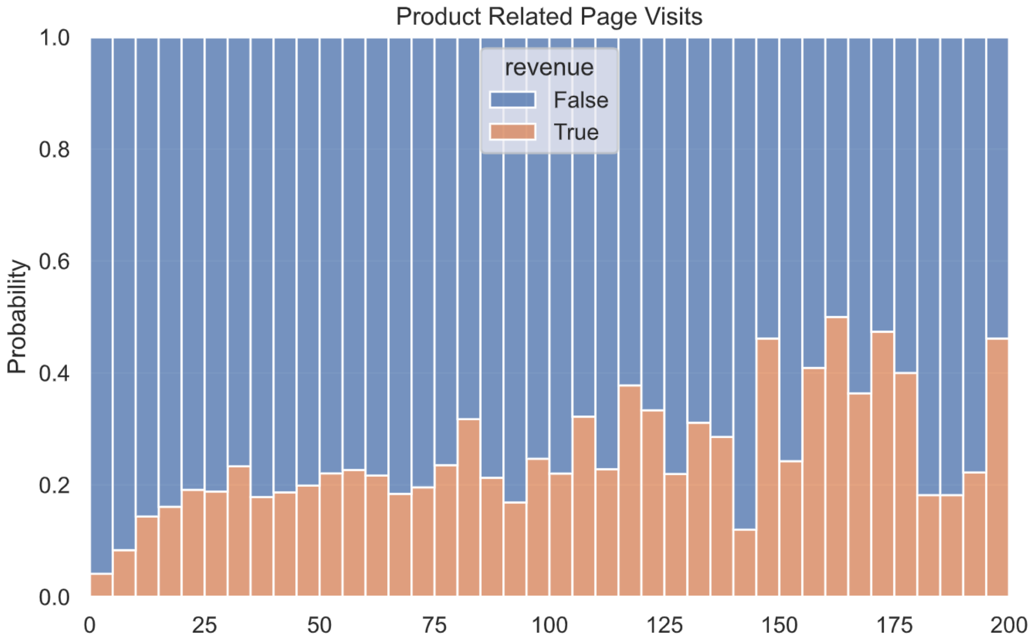




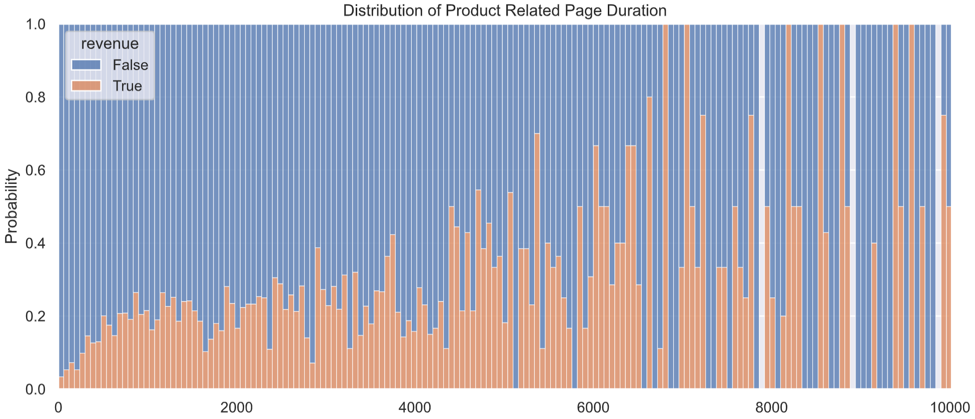
In the probability charts above we clearly note that the likelihood of a purchase increases as page value increases. This variable has a big impact on our target class.

Product Related Pages And Duration. First, we'll look at the number of pages visited, then the duration of the visit, and finally we'll explore how each interacts with our target class.

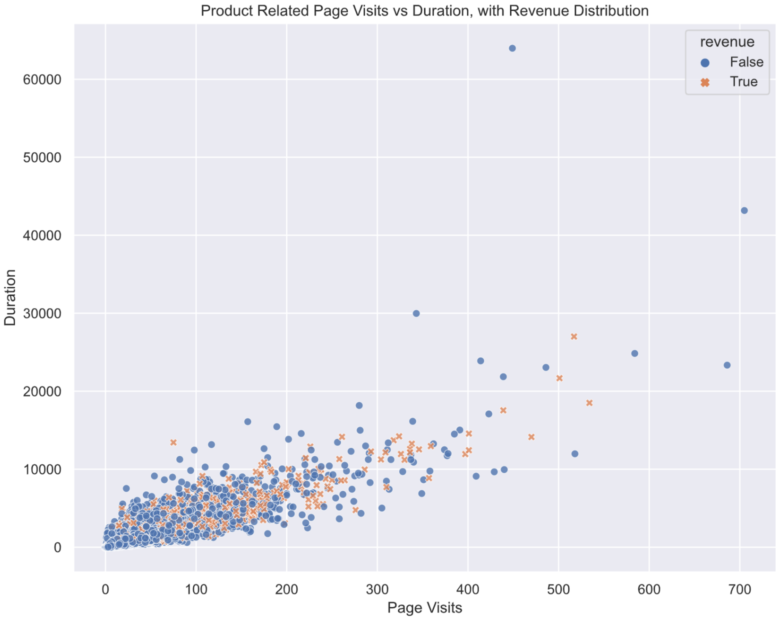
The product-related page visits probability histogram below shows that there is an increasing trend for the probability of revenue generation as the number of product-related page visits increases.



For the duration of product-related page visits, unfortunately, our reliable probability plot provides us with less insight, as the sample size gets much smaller when the duration increases.

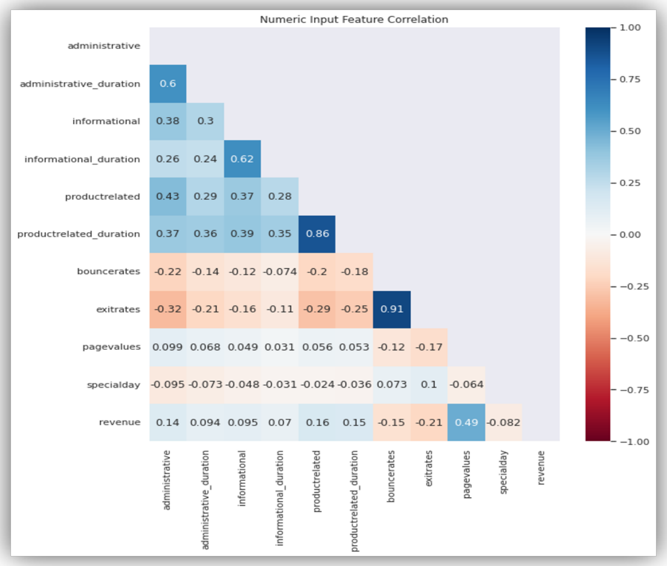


To resolve this issue we combined product-related page visits and duration into a scatter plot to better evaluate the effects of these features on revenue. While we noted that individually, product-related page visit frequency and product-related page duration seemed to have a positive effect on revenue distribution, this becomes harder to recognize when looking at the scatterplot.

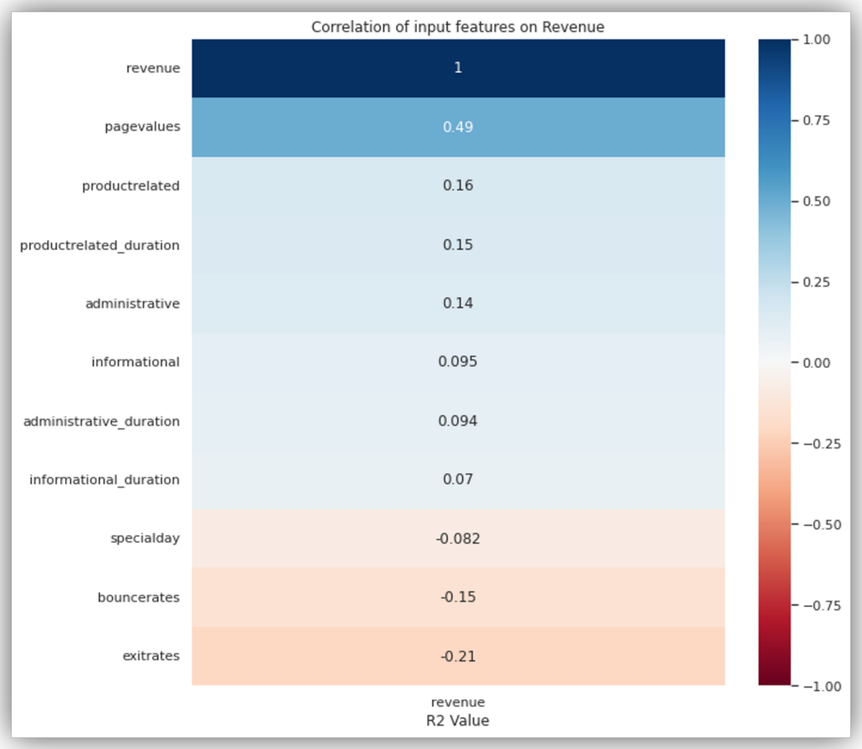


After creating a fitted model, we will evaluate feature importance to get a better understanding of these features.

Correlation Heatmap



The correlation between the numeric variables was analyzed with the aid of correlation heatmaps. The second correlation coefficient chart below clearly lists the correlation coefficients between the target class “revenue” and each numerical input feature in descending order. The numeric feature “pagevalues” has the strongest positive correlation with “revenue” (r=0.49), followed by “Product related page visits” (r=0.16) and “productrelated_Duration” (r=0.15); the feature “exit rates” is negatively correlated with “revenue” (r= -0.21) , followed by “bounce rates” (r=-0.15) . In contrast, informational page visits/duration, administrative duration, and the proximity to holidays do not have a strong correlation with “revenue”.



Predictive Analysis and Modeling

As mentioned previously, predictive analysis is primarily completed with the use of scikit-learn and the imbalance-learn libraries. After performing EDA to gather initial insights about potential patterns that exist within the dataset, we perform some basic data preparation steps and configure our modeling framework.

At the core of our modeling is the Pipeline tool provided by Scikit-Learn. Using it allows us to reduce potential data leakage, improve the uniformity of the input features, and provide a strong foundation for iterative experimentation with greater efficiency. To use it, we first created two pre-processing steps:

Pre-Processor:

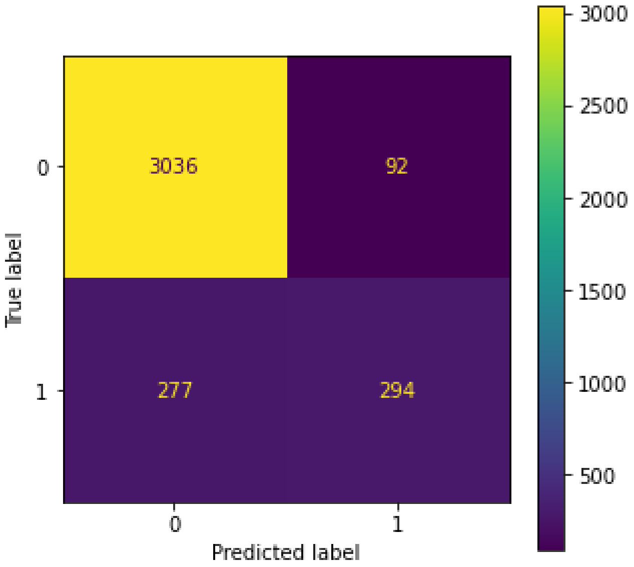
1. Numeric Transformer: StandardScaler() to normalize all numeric features
2. Categorical Transformer: uses OneHotEncoder() to encode categorical features.

We combine these two items into a single “pre-processing” step that is applied to our data and then feeds that transformed data into the model for fitting and predictions. To summarize, our base modeling structure is as follows(it can also be viewed in the accompanying notebook for a more detailed understanding):

Pipeline:

1. Step 1: Pre-Processor(normalizes numeric, encodes categories)
2. Step 2: Estimator(transformed data from step 1 fitted to model)

Utilizing this framework allows us to easily incorporate grid search and other tuning and pre-processing techniques to optimize a given estimator for our use case. In this initial iteration of modeling, we use a RandomForestClassifier to test the preliminary results of our data. This initial framework provides us with the following results:



Recall that in our use case, ‘0’ represents no revenue, and ‘1’ represents revenue. As we can see from the confusion matrix, the model performs well when identifying activity that does not result in revenue, however, it performs virtually no better than a coin toss when predicting “revenue”. We’re primarily interested in improving revenue label identification because this supports our stated goal of identifying consumers likely to make a purchase, in order to take steps to ensure this transaction takes place. First, we need to overcome the obvious class imbalance.

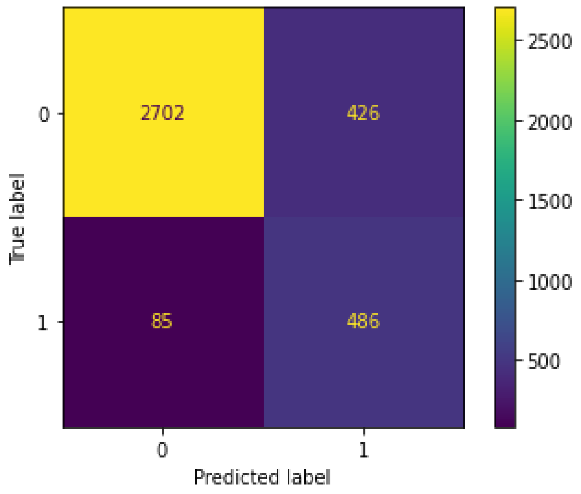
A common technique to handle imbalanced target classes is a combination of oversampling and undersampling - both of which can be accomplished using functions found in the ImBalance Learn library. Since we’ve already configured our ML use-case to use pipelines, we can easily implement these tools into the preprocessing steps of our ML implementation. We do this by adding two more steps to our existing pipeline. These steps are an oversampling technique called “SMOTE” that synthetically adds more instances of our class ‘1’, and a random undersampling step that randomly eliminates instances of our majority class. Together these two steps provide the model with training that contains a more balanced target feature.

<u>Pipeline:</u>

1. Pre-Processor(normalizes numeric, encodes categories)

2. Oversample(apply SMOTE to minority class)
3. Undersample(reduce instances of majority class)
4. Estimator(transformed data from step 1 fitted to model)

This specific combination of undersampling and oversampling techniques was chosen as a result of previous research that concludes this specific combination often yields strong results (Chawla et al., 2002). The end result is that target classes are more evenly balanced and this allows the model to predict labels correctly with much greater accuracy.

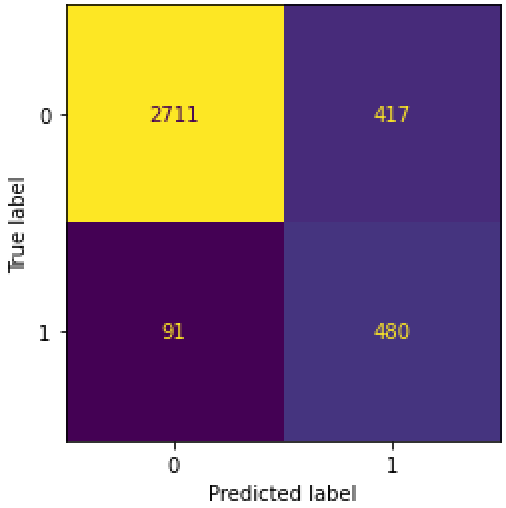


We note that the False Positive predictions have decreased substantially, and the model can more frequently predict a correct label of "Revenue: True".

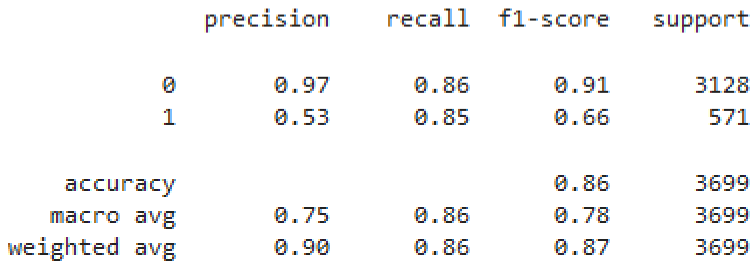
In the interest of a comprehensive analysis, we also go on to evaluate all available classifiers. To identify the best classification estimator for our use case, we utilize GridSearchCV's ability to optimize for the desired outcome. This provides us with the ability to select 'recall' as a scoring metric, which will provide us with the best performing model with the ability to identify class '1'. Note that in our accompanying notebook, we check whether "f1" or "recall" weighted scoring provides the best outcome - we find that in our use case, selecting "recall" provides us with a model that can predict class "1" most reliably. We checked all possible classification estimators(there are 41!) that Scikit-Learn provides, and then told GridsearchCV to find the best model for our use case. Based on optimizing for recall, a grid search identifies the following three models as optimal for our use case:

1. RandomForestClassifier() - Recall Score: 85.4%
2. GradientBoostingClassifier() - Recall Score: 85.4%
3. AdaBoostClassifier() - Recall Score: 83.8%

By pure chance, we've already chosen RandomForestClassifer as our baseline model, but, We'll double-check on GradientBoostingClassifier just to ensure that we don't leave potential "performance" on the table. The confusion matrix for GradientBoosting confirms that we have already selected the optimal model for our use case. We can see that while this type of model still performs much better than our first iteration, it has a slightly higher rate of error in terms of type 1 and type 2 error rates. As we've previously discussed, we want to tune our model to identify as many cases of class "1" as we can - so we'll switch back to our RandomForestClassifier as our final model choice.



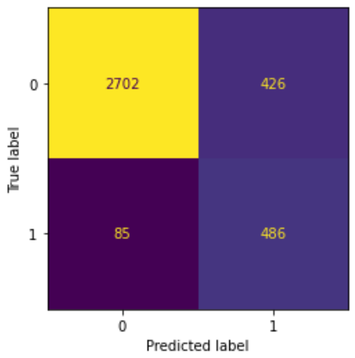
Scikit-Learn's random forest classifier is an ensemble model its page entry describes as "After many decision tree classifiers have been fitted to various dataset subsamples, a random forest is a meta estimator that uses averaging to improve predicted accuracy and decrease overfitting." (Buitinc et al., n.d.). Because of the way that this estimator was built, we thankfully don't have to take many steps to optimize its performance. It does have a variety of parameters that we can elect to tune, including the ability to tune class weights which we also explored. Let's review the scoring for our RandomForest model quickly. We get the following classification report and confusion matrix:



```
              precision    recall  f1-score   support

           0       0.97      0.86      0.91      3128
           1       0.53      0.85      0.66       571

    accuracy                           0.86      3699
   macro avg       0.75      0.86      0.78      3699
weighted avg       0.90      0.86      0.87      3699
```



From these, we can see that the recall score for both classes is virtually the same, which indicates that we might not be able to get much more performance from our model. While we won't show each of the steps here, we did check for other ways to improve our base mode. We added a recursive feature elimination step to our pipeline that automatically selected the ideal number of input features, but this failed to score as well as our already existing model. Tuning hyper-parameters listed on the scikit-learn documentation for the estimator also failed to produce a higher scoring model, in spite of the fact that we dedicated 22 hours of computing time, which considered 9.6 million different combinations of parameter values. In fact, the only technique that improved our model performance was tuning class weights. By reducing the weight of class '0' and increasing the weight of class '1', we were able to add one more correct prediction. While one extra prediction will have little to no impact on the scale of the average e-commerce site, we always want to optimize our model as much as possible. As a reminder, our application of each of these techniques can be further explored in the accompanying notebook.

To summarize our modeling application, we utilize a pipeline to make iterative changes more easily while reducing data leakage and other opportunities for error. After testing every available classifier in sklearn, we reach the conclusion that RandomForestClassifier is best suited for our business use case. After carefully tuning and optimizing for predicting class 1 we were able to produce an ML application that can correctly predict if a customer will make a purchase with

85% confidence - and when a customer won’t make a purchase with 86%. We find this acceptable in our business context, due to the many other factors that influence a customer's decision-making that are simply impossible to capture, like sentiment, mood, or any other of a limitless number of reasons external to web browsing patterns that affect human decision-making. We feel that this ML application provides a very capable platform that can be used by other departments of an e-commerce site to use for more advanced techniques that increase consumer retention and probability of purchase.

CONCLUSION

We've successfully demonstrated that ML can provide a strong foundation for improved decision-making and impact metrics. From a business perspective, machine learning can be exploited to identify patterns in consumer purchasing behavior, improve market understanding, and support dynamic pricing and purchase incentive programs. From a consumer retention perspective, ML can be used to drive marketing efforts that drive personalized interactions with storefronts, and aid companies in optimizing website design to maximize the likelihood of a purchase.

REFERENCES

Olenski, S. (2017, February 21). *6 reasons your website visitors are not turning into customers*. Forbes. Retrieved August 8, 2022, from https://www.forbes.com/sites/steveolenski/2017/02/20/6-reasons-your-website-visitors-are-not-turning-into-customers/?sh=68a783b52f33

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Couto, J., Fagioli, M., & Umpierrez, G. (2022, March 21). *The Guide to Machine Learning in Retail*. Tryolabs. Retrieved August 17, 2022, from https://tryolabs.com/guides/retail-innovations-machine-learning

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013, September 1). *API design for Machine Learning Software: Experiences from the scikit-learn project*. arXiv.org. Retrieved August 17, 2022, from https://arxiv.org/abs/1309.0238

Y.Ku ,Y.Tai.(2013).*What Happens When Recommendation System Meets Reputation System*?

Requena, B., Cassani, G., Tagliabue, J., Greco, C., & Lacasa, L. (2020). Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Scientific Reports*, *10*(1), 16983. https://doi.org/10.1038/s41598-020-73622-y

Coppola, D. (2021, October 13). *Digital buyers worldwide 2021*. Statista. Retrieved August 17, 2022, from https://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide/

Mohsin, M. (2021, June 20). *10 Online Shopping Statistics You Need to Know in 2021 [Infographic]*. https://www.oberlo.com/blog/online-shopping-statistics

Soper, T. (2020, April 7). *COVID-19 crisis sparks 'inflection point' for online grocery—And huge revenue for Amazon*. GeekWire. https://www.geekwire.com/2020/analyst-covid-19-crisis-sparks-inflection-point-online-grocery-huge-revenue-amazon/