Real-time Prediction of Online Shoppers' Purchasing Intention Using K-Nearest Neighbor Model

Pranav Tonpe

ABSTRACT

[Objective] With the downturn of in-person shopping after the ramifications of the Covid-19 pandemic, major retailers like Amazon and eBay have taken the spotlight in this new age of online retailing. Online retailing, more widely known as E-Commerce, is characterized by the ability of customers to search, select, and purchase certain products remotely over the Internet. The main issue revolving around online shopping lies in the misclassification of user intent on any particular website leading to irrelevant content which has exacerbated customer satisfaction. Therefore, the purpose of this project is to develop a highly accurate machine learning model with maximized true positive and negative rates in hopes of correctly identifying a user's purchasing intent

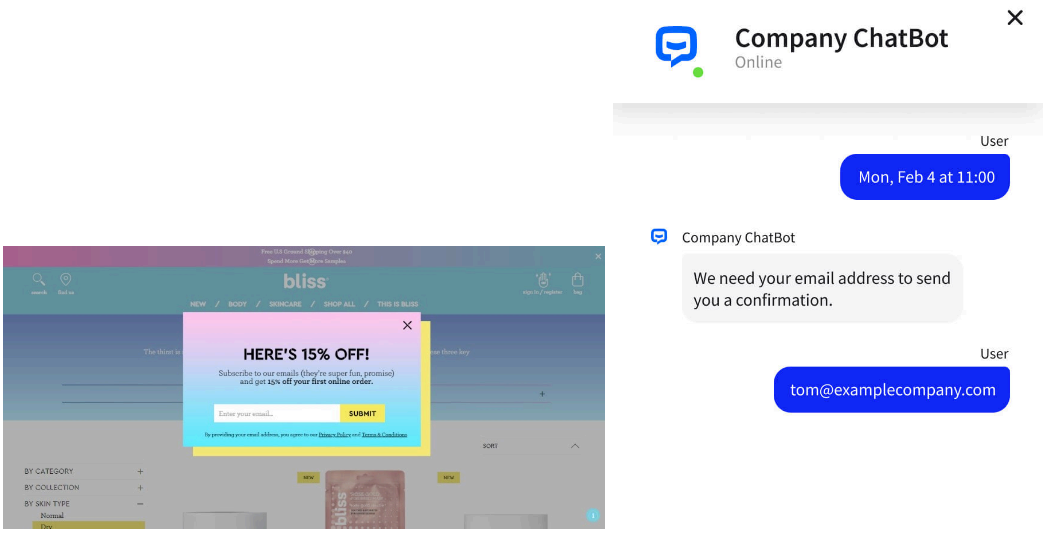
[Methodology] A machine learning software was created using K-Nearest Neighbor classification which identifies key features such as bounce rates, exit rates, and duration of the session as well as the type of page visited (Administrative, Product Related, or Informational) to analyze consumer behavior, specifically their likelihood to purchase a product.

[Results] With a 90-10 train-test split, the model had approximately an 89% true positive rate which is the proportion of users who did go ahead with a purchase and were correctly identified while a 43.6% true negative rate characterized by those who did not purchase and were correctly identified by this model. Throughout numerous test cases, out of a sample size of slightly over 1200, the model correctly predicted around 1020 users with their buying intention while incorrectly identifying around 212 users.

[Conclusion] In the future, this low-cost classification tool can be implemented in higher-level marketing and advertising agencies trying to maximize user participation by personalizing websites based on a user's actions. This application can be altered to include a wide variety of website analytical features most commonly seen with Google's advertising campaigning. However, this developing application serves as an effective instrument for examining the intent of online shoppers with much room for improvement.

INTRODUCTION

The world of e-commerce impacts millions of people worldwide with a single tap of a button on their phones indulging a user in an attention-seeking environment. With modern-day smartphones advancing yearly, data processing and storage capabilities have allowed for cutting-edge web creations that capture the attention of online shoppers in hopes of promoting a positive emotional response from the user. This is exactly where Artificial Intelligence comes in and its implications have certain benefits and drawbacks. In fact, complex machine learning algorithms utilize behavioral and transactional data to enhance an understanding of a customer's needs[1]. Ideally, e-commerce retailers can transform this data into a personalized experience with user-oriented shopping recommendations in real-time. More importantly, retaining customers online with thousands of retailers waiting at their doorstep is a crucial step toward maximizing profits. Recent studies show that retargeting website visitors increases the likelihood to buy by 43% with 92% of online shoppers not buying a product upon their first visit[2]. This idea of reminding past customers of some interaction they had on their website is effective in keeping users on company radars and it opens the door to future possible shopping ventures. However, the issue lies in the early identification of a user's intent because it is challenging to completely understand human behavior especially when given a plethora of choices. Oftentimes, users are misled by computer algorithms because of a failure to realize that a user may be visiting a website for informational rather than retail purposes for example. When presented with a live chat box or even a flashy discount advertisement in the middle of reading a lengthy article, users may find it bothersome to be distracted by such media. As a matter of fact, over 70% of internet users automatically disapprove of pop-up ads while ad-blocking software rose in popularity to over 200 million daily active users[4]. This ultimately traces back to the lack of apprehension between major online retailers and potential customers. By accurately analyzing patterns or trends in customer behavior right from the moment the website loads onto the screen, marketing agencies can bolster their conversion rate (percentage of visitors that land on a website and complete the desired action) and begin to run more engaging, cordial campaigns in the future[3].



Mon, Feb 4 at 11:00

We need your email address to send you a confirmation.

tom@examplecompany.com

HERE'S 15% OFF!

Subscribe to our emails (they're super fun, promise) and get 15% off your first online order.

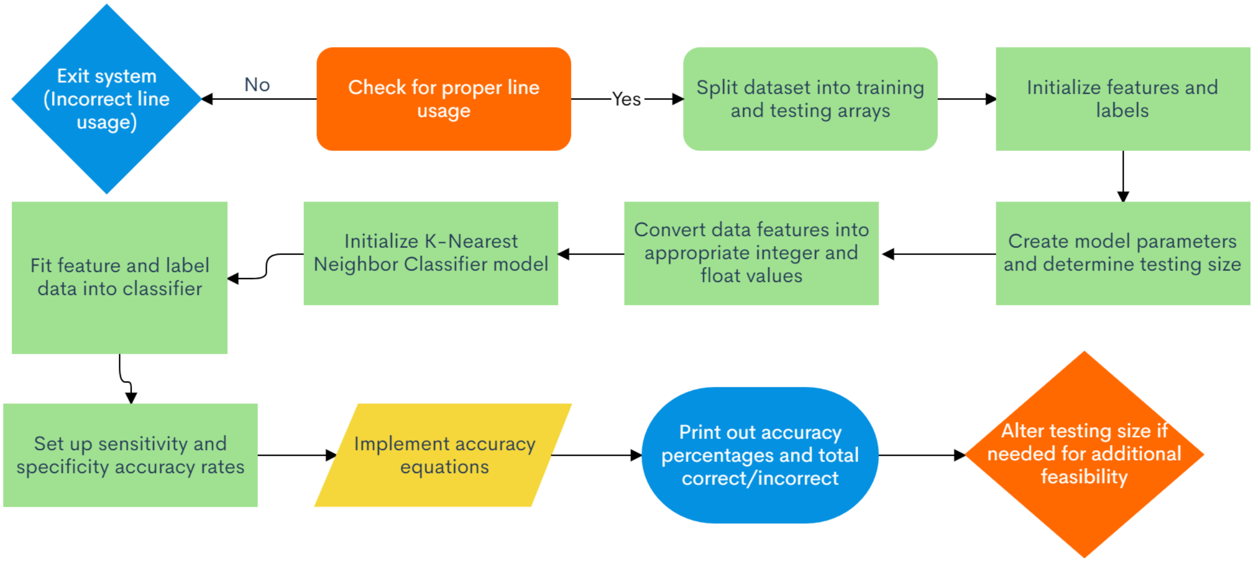
Enter your email...

By providing your email address, you agree to our Privacy Policy and Terms & Conditions

(Figure 1.1, 1.2) Pictured are two examples of personalized media in the form of pop-up ads and chat box robots as a result of AI algorithms

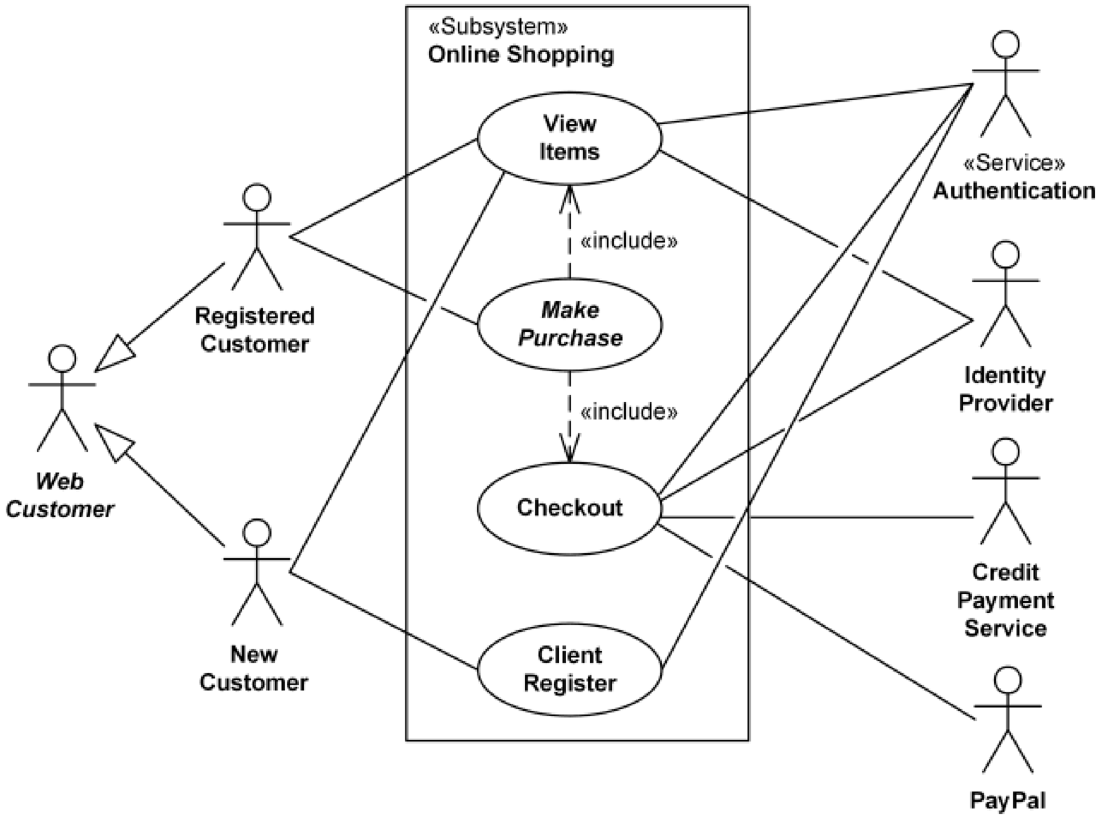
METHODOLOGY

The construction of this application is separated into 4 main sections: data processing, train-test setup, feature initialization, and data fitting before running the code on a local device. Data processing includes loading the data from an external source to a personal hard drive. Additionally, the features must be identified in the large comma-separated values file which was initially converted into a Microsoft Excel spreadsheet for visual purposes. For testing, command-line arguments ensured proper usage. The train-test setup includes a function from the sklearn model selection library which essentially splits a certain amount of data into X and Y 2-dimensional arrays that can be used later for the nearest-neighbor model and for arbitrary testing. Feature initialization focuses on converting dataset values into appropriate floats or integer variables so they can be implemented in the classifier. For example, the browser variable may return 1 of 3 numbers indicating whether a potential shopper is using Google Chrome, Firefox, or Internet Explorer for instance. Similarly, the weekend variable was converted to a boolean which returned either a true for when it was a weekend or false for otherwise. Finally, the data is fitted into the K-Nearest Neighbor classifier by inputting both the evidence and the labels respectively. Before running the program, a specificity and a sensitivity variable are initialized to evaluate the effectiveness of the software. These accuracy metrics are then converted into a contiguous flattened array that follows a simple arithmetic function that divides the true positive rate with the sum of the true positive and the false negative rates [5]. Correct and incorrect predictions are calculated and counted using a confusion matrix that tests for precision.


Exit system (Incorrect line usage)
No
Check for proper line usage
Yes
Split dataset into training and testing arrays
Initialize features and labels
Create model parameters and determine testing size
Convert data features into appropriate integer and float values
Initialize K-Nearest Neighbor Classifier model
Fit feature and label data into classifier
Set up sensitivity and specificity accuracy rates
Implement accuracy equations
Print out accuracy percentages and total correct/incorrect
Alter testing size if needed for additional feasibility

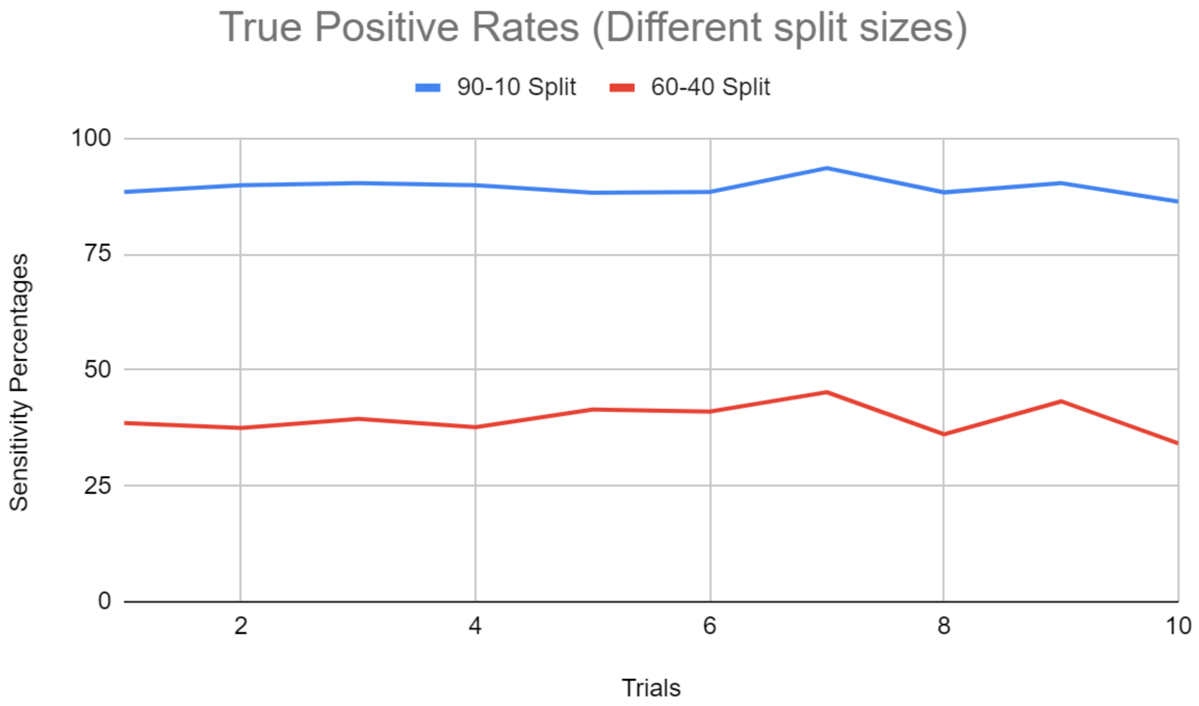
(Figure 2) Classifier model process

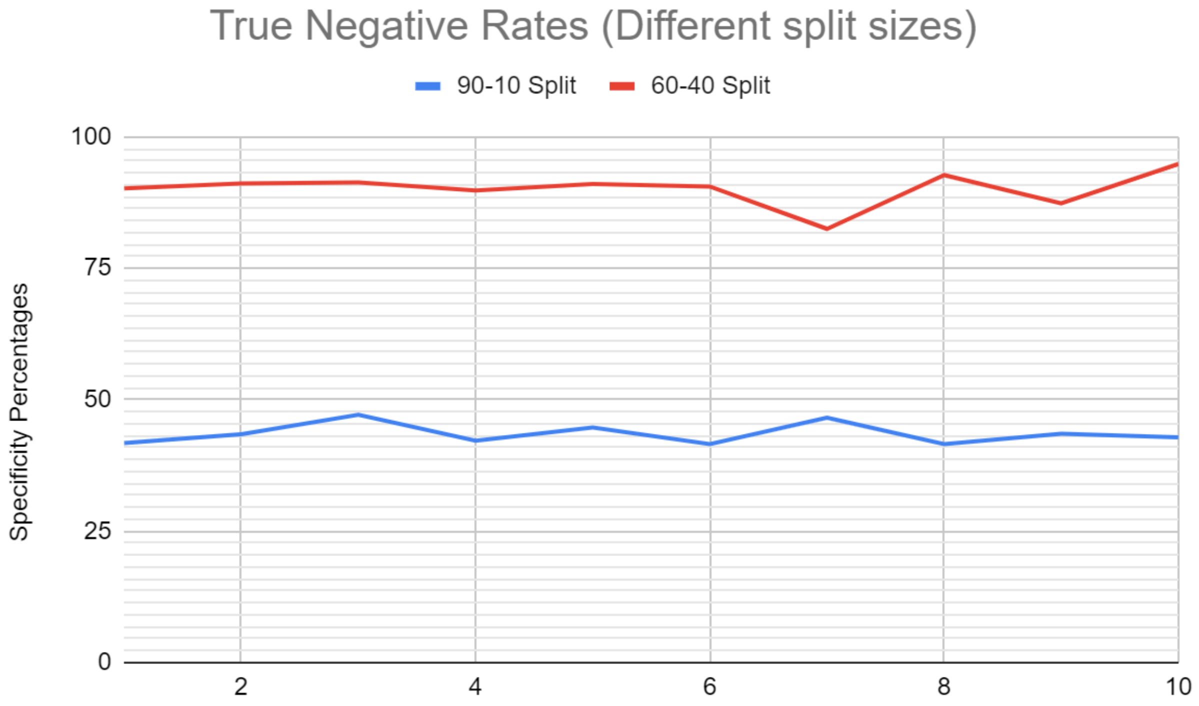
The key distinction between returning visitors and new visitors is highly influential in the placement of a user's intent. After identifying the type of visitor and several distinct user information like the operating system, traffic type, and region specification, the model proceeds to classify which users would be more inclined to purchase an item from a website.



(Figure 3) Differentiation between new and returning visitors and hypothetical steps to purchase[6]

RESULTS

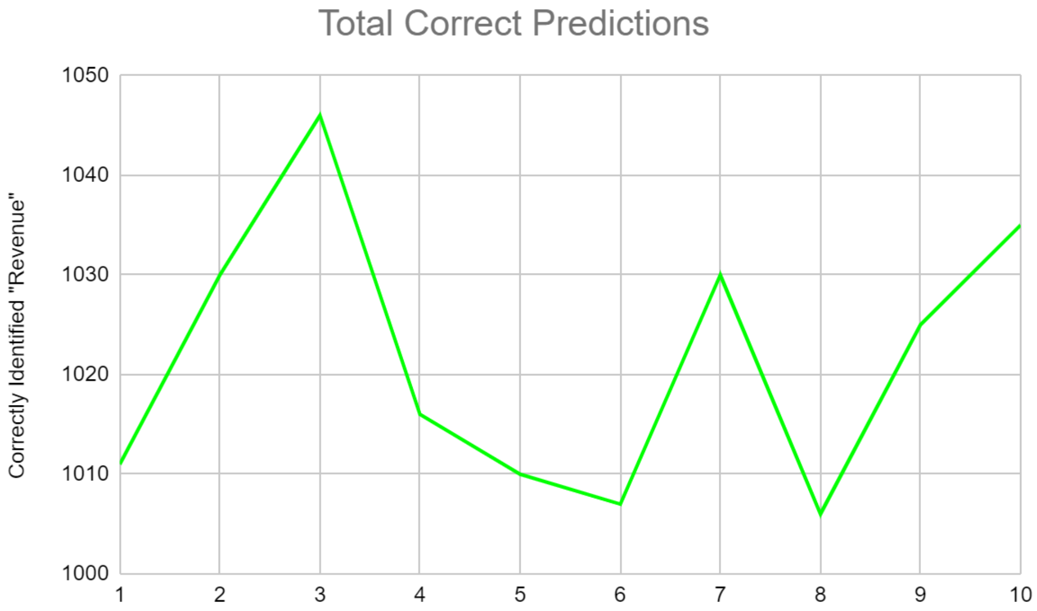

True Positive Rates (Different split sizes)
90-10 Split
60-40 Split
Sensitivity Percentages
100
75
50
25
0
2
4
6
8
10
Trials

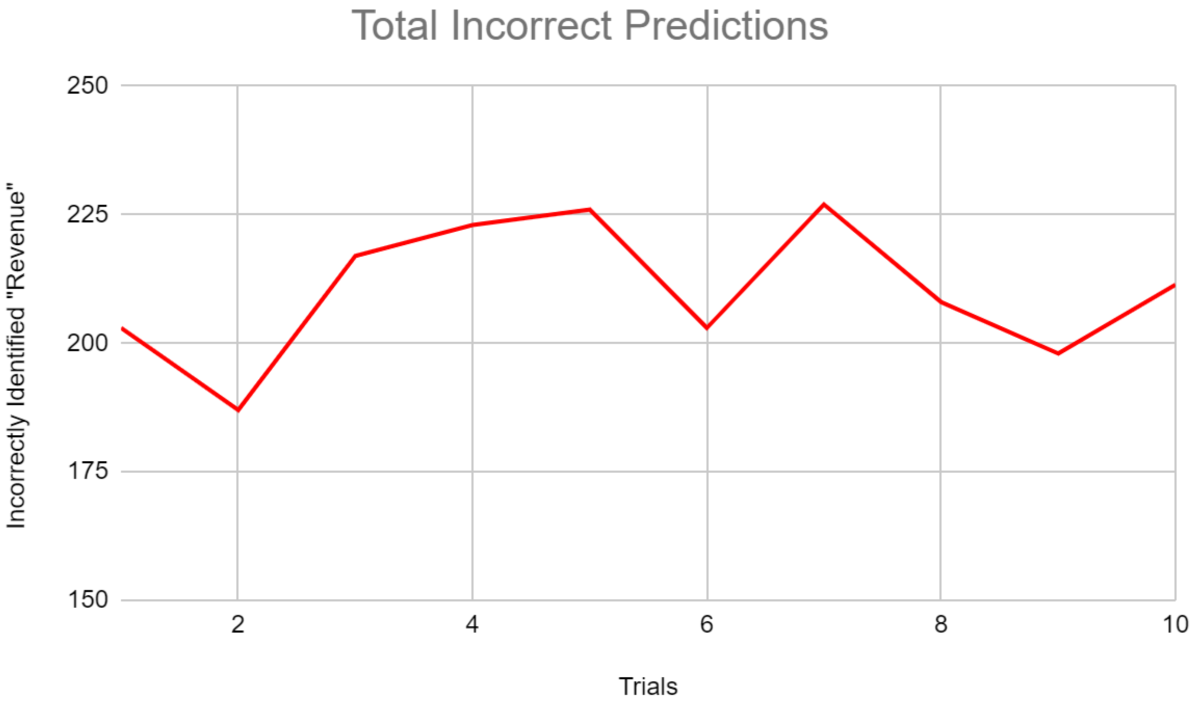

True Negative Rates (Different split sizes)
90-10 Split
60-40 Split
Specificity Percentages
100
75
50
25
0
2
4
6
8
10

(Figure 4.1, 4.2) Data collected over 10 randomized data trials

[90-10 Split] For 9/10 (90%) of the trials, the classifier model was able to accurately predict user intention based on the given dataset with over 88% accuracy with one outlier being around 86%. Across each trial, the sensitivity percentages did not fluctuate to a large degree which may show the software's consistency. However, there were several spikes and dips in the data graph which may have been a result of the random data sampling used. On the other hand, the true negative rates or the specificity accuracy metric fared out to be around 40-45% accurate. In a testing sample size of around 1200, the classifier was able to correctly identify 1020 user intent on average with around 210 incorrect predictions. 30% of the sensitivity percentages based on trial results crossed the 90% mark with nearly 1050 correctly identified user intent while 10% of the trials or 1/10 went below the mean sensitivity percentage of 89.376% with an accuracy of 86.37%. In 100% of the specificity trials, the percentages were over 40% accurate with very subtle differences throughout test cases.

[60-40 Split] One interesting observation made was how 100% of the sensitivity percentages dropped when the testing size was increased to 40% rather than 10%. From an average of 89% in the 90-10 split, the sensitivity rate dropped to a whopping 39.44% possibly due to the increased accessibility of data when training and testing. The specificity rates actually increased from an average of 43% to about 90%. With a much larger sample size, the classifier model was able to accurately predict around 4050 user intentions while incorrectly predicting around 860. By evaluating the instances when a false negative or a false positive prediction occurs, a higher sample size will affect the sensitivity and specificity accordingly.


Total Correct Predictions
Correctly Identified "Revenue"
1050
1040
1030
1020
1010
1000
1
2
3
4
5
6
7
8
9
10


Total Incorrect Predictions
Incorrectly Identified "Revenue"
250
225
200
175
150
2
4
6
8
10
Trials

ACKNOWLEDGEMENTS

I would like to thank C. Okan Sakar and S. Olcay Palat for providing this easy-to-use shopping dataset with definitions for key features and labels. I would like to thank all the AYSI officers who made this summer coding institute a wonderful learning experience even in a shortened time. Even with less time for the competition, I am thankful for this opportunity to implement an application of my own. Although there were many struggles throughout this journey, I hope to take these efforts in my future of artificial intelligence. Finally, I would like to thank my family and friends for their constant support.

REFERENCES

[1] "Using AI to create a personalized shopping experience in online retail." 24 Jun. 2020, https://dmexco.com/stories/using-ai-to-create-a-personalized-shopping-experience-in-online-retail/. Accessed 22 Jul. 2023.

[2] "15 Eye-Opening Online Shopping Statistics for 2021 - Sleeknote." 15 Apr. 2021, https://sleeknote.com/blog/online-shopping-statistics. Accessed 22 Jul. 2023.

[3] "Machine Learning For Ecommerce: How Does it ... - BigCommerce." https://www.bigcommerce.com/blog/ecommerce-machine-learning/. Accessed 22 Jul. 2023.

[4] "Do Pop-up Ads Actually Work? Here's the Data You Need." 10 Jun. 2016, https://www.smartbugmedia.com/blog/do-pop-up-ads-actually-work-heres-the-data-you-need. Accessed 23 Jul. 2023.

[5] "numpy.ravel — NumPy v1.21 Manual." https://numpy.org/doc/stable/reference/generated/numpy.ravel.html. Accessed 24 Jul. 2023.

[6] "Online shopping UML examples - use cases ... - UML-Diagrams.org." https://www.uml-diagrams.org/examples/online-shopping-example.html. Accessed 24 Jul. 2023.