Comparison of Techniques to predict Shopper's Intent

[1]Harsh Salvi, [2]Nemil Shah, [3]Tilak Satra, [*]Prof. Mitchell D'Silva

[1,2,3]Student, [*]Faculty
Department of Information Technology,
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India.

Abstract : The burgeoning growth of the internet these days has resulted in a spurt in online business. Online E-commerce websites seek to convert the potential visitors on the websites to buyers of the goods and services. The behavior of the customers browsing online is used to predict the intent of an online buyer dynamically. The anonymous nature of these e-commerce transactions presents a greater risk for the industry with regard to increasing sales, enhancing customer experience, etc. as it becomes difficult to access the customer's browsing patterns. Such patterns can show customer's needs, expectations, and dislikes. The user's aim can be predicted early in the browsing session based on the moments tracked by the task performed by the user on the website. There are various dynamic predictive models such as Naive Bayes, Random Forest, XGBoost, KNN, Logistic Regression that are used for predicting user's intent before they would leave the website. The performance of these models has been examined.

***IndexTerms* - Online shopper intent prediction, intention prediction, logistic regression, KNN, Random Forest, Naive Bayes, XGBoost.**

I. Introduction

A success for an e-commerce website is measured through its efficient use of business models and the loyalty of customers [1]. The importance of customer loyalty is given crucial weight because the customers on the internet are fickle and volatile. The customers get everything at their fingertips which has motivated a large crowd towards online shopping. They may move on to other websites as soon as they find better options available elsewhere. A user's intention of performing some particular task depends upon a goal they are trying to achieve by connecting to that specific website [3]. If the product or the services provided by the business or the website satisfies user's demand such as product availability, least cost, best quality, etc. or else the customer moves to another website for meeting their needs. A consumer's click path can be analyzed to predict if the consumer will help in generating revenue [4]. This paper aims to compare five machine learning algorithms for predicting customer behavior.

II. Dataset Description

The dataset used for determining a customer's intent is the Session Database which consists of 12,330 sessions or entries. The dataset consists of 10 numerical and 8 categorical attributes. The Page Value attribute for a particular web page tells about the number of customers that visited the page before performing a transaction on the website. The Special Day attribute showcases the probability of the customer transacting from the website on special occasions like Children's Day or during Sale Season and so on. Other attributes like Administrative, Administrative Duration, Informational, Informational Duration, Product Related and Product Related Duration attributes represent the number of different types of pages visited by the customer in a particular session and also the total time spent on each of the pages. The Bounce Rate attribute determines the percentage of customers that visit the site through a particular web page but do not perform any functions and thus leave the website. The Exit Rate attribute gives the percentage of customers who visited a particular web-page or the site.

III. Methodology

The first step of the implementation is to split the dataset for training and testing [5]. The dataset is pre-cleaned and hence can directly be used. A standard split ratio of 70:30 is used. This results in 8631 training samples and 3699 testing samples. Once the datasets are ready, the models need to be trained. This paper studies the implementation of five different classification algorithms: Random Forest, Logistic Regression, XGBoost, Naive Bayes and K Nearest Neighbors on the problem of online shopper's intention prediction [2].

Decision trees are the centerpiece of the Random Forest algorithm. Decision trees are pretty intuitive. They consist of multiple conditions and according to those conditions, different branches are formed. These branches lead to the final class for any tuple. Each condition can lead to two or more branches. At each node, the condition and the branches are to be decided. Random forest is used to randomly create a forest of different decision trees. There is a direct correlation between the number of random trees created and accuracy. Random forest algorithm has been implemented using the sklearn library.

Logistic regression, which is one of the most basic classification algorithms, is used as a starter algorithm. It is a statistical model that uses the standard logistic function. The output of the logistic function gives a range of 0 to 1. A threshold of 0.5 is set to convert the output to the classification label. 0.5 is the standard value of threshold used to divide the output range into two equal parts. The optimization algorithm used is the Limited-Memory Broyden–Fletcher–Goldfarb–Shanno (Limited Memory BFGS) algorithm. This algorithm tries to get values closed to the standard BFGS algorithm using limited memory. BFGS is an iterative method used for finding the parameters of the model. It belongs to the hill-climbing set of algorithms. This model also uses a regularization technique. The regularization technique used is the L2 regularization. It adds a penalty that is equal to the square of the magnitude of all coefficients. Regularization is done to limit the model from overfitting. With the kind of data present, the chances of a model overfitting are very high. This makes regularization very important in this case. The model was implemented using the sklearn library.

XGBoost offers high-performance implementation of gradient boosted decision trees. The XGBoost library has an underlying C++ codebase with a python interface. Boosting has an iterative process. It does not use the conventional ensemble technique. Instead of training all the models in isolation, the models are trained in succession. Each model is trained in such a way that it corrects the mistakes made by the previous one. The advantage of such a training method is that the models are more focused on correcting the mistakes and unlike traditional methods, it does not allow the mistakes to be repeated across models. In the gradient boosting approach, new models are trained to predict the errors of the previous models before being added to the ensemble. Fig 1 below outlines the process. The default learning rate of 0.3 and the default maximum depth of 6 is used. The tree model used is the approximate greedy model which is suitable for medium-sized datasets like the one being used.

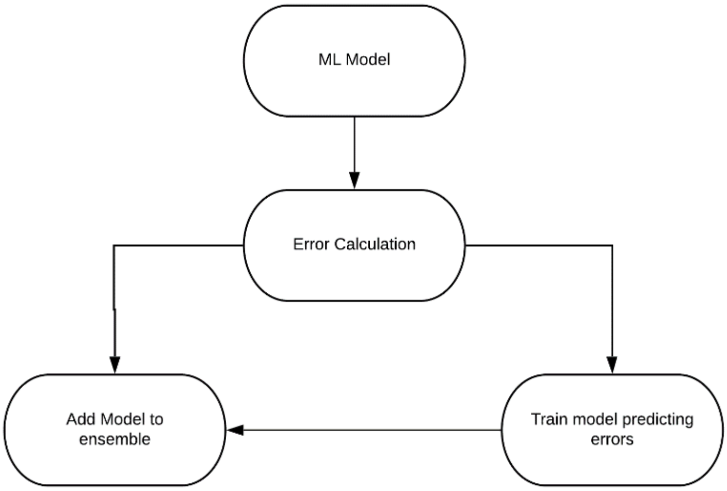

Fig 1. XGBoost Process

Naive Bayes is a probabilistic algorithm. It is based on the concept of Bayes theorem. Bayes theorem describes the probability of an event occurring based on the occurrence of other events. This type of probability is termed as a conditional probability. Bayes theorem is used to find mathematical relations between conditional probabilities. The Naive Bayes algorithm generates the probability of a tuple belonging to a particular class (y) based on the probability of existing conditions (x). The Naive Bayes algorithm relies on a presumption that every feature (every column in x) is independent of one another. This, however, is not true for most datasets including this dataset. This leads to a loss of information by the model. The probability of an event occurring is also calculated based on the number of occurrences in the dataset. This leads to a very inaccurate model when the dataset does not represent the real-world scenario. Naive Bayes is still a very fast algorithm and is used as a starter algorithm for many classification problems. Naive Bayes implementation is done using the sklearn library.

K nearest neighbors is a non-parametric classification algorithm. It is based on the simple concept that similar things are near to each other. KNN does not make any assumptions about the data. This implementation uses a K value of 5. This means the five nearest neighbors of any test data point are found out. The majority class of the nearest neighbors is then selected as the class for the test tuple. The biggest difficulty in using KNN is finding out the value of k. The K value of 5 is used as it gives the best result in this case. However, it has to be noted that this accuracy very much depends upon how the data is split. This means that the K value of 5 may not be the ideal value for the real-life implementation of the algorithm. The KNN algorithm is also not scalable and time-consuming. For each testing tuple, it needs to find the distance from every training tuple to find the nearest neighbors. KNN is also more useful when the data is continuous rather than categorical. Measuring distances for categorical data loses the information contained in the category. It also requires the categories to be arranged in a particular order. KNN, however, is an easy starter algorithm for small to medium-sized datasets like this one.

IV. Results

Table 1. Results

Algorithm	Precision	Recall	F1-Score
Logistic Regression	0.74	0.35	0.47
Random Forest	0.74	0.50	0.59
XGBoost	0.74	0.58	0.65
Naïve Bayes	0.42	0.61	0.50
KNN	0.64	0.28	0.39

Table 1 above shows the results of implementing the various algorithms. Precision is the score which indicates how accurate the model is based on the positives predicted. It is the “ratio of how many times did the customer-generated revenue to the number of times model predicted it would”. The recall value is used to calculate how many of the positives were predicted out of the total. The recall is the “ratio of how many times the model predicted the customer-generated revenue to the actual number of the times the product generates revenue”. F1-score is the harmonic mean of precision and recall. It is used as a middle-ground metric to compare the algorithms. As can be seen from the results above, it is easier for all the algorithms to score a higher precision score rather than a recall score. This is because of the highly skewed data. With only a few examples of revenue being generated, the algorithms find it difficult to predict the cases when revenue is actually generated.

The F1-score is highest for the XGBoost algorithm followed closely by the Random Forest algorithm. The close values are due to the fact that both the algorithms use a tree classifier. While Random Forest randomly generates decision trees, XGBoost uses a more systematic method to generate decision trees. This gives it the upper edge. The F1-score for XGBoost can be further improved by tuning various parameters. These results prove that tree-based classifiers are better suited for such a problem.

Naive Bayes has the best recall score for the dataset. However, it significantly suffers in the precision department. This is because the Naive Bayes algorithm considers all columns in a dataset to be independent. This leads to it correctly finding the most effective columns and a higher recall score. However, due to it considering columns to be independent, it ignores the information about relations between different dependent columns. This leads to a poorer precision as it is more likely to label a tuple as positive based on only a few columns and ignoring the rest of the columns.

K Nearest Neighbors algorithm performs the worst in case of the recall metric. This is because KNN works better with numeric continuous data rather than categorical data. Categorical data loses its meaning when used in a distance-based algorithm. With no proper numeric data given to the categories, they lose meaning when using the KNN algorithm. Greater the value of other numeric data, the lesser the importance is given to categorical data. This effect worsens when using a skewed dataset like this one. Since this dataset contains many categorical data, the KNN algorithm has a hard time finding out the times when a customer actually generated revenue.

V. Conclusion and Future Work

The behavior of the visitors based on their actions or browsing pattern is important for various businesses as it helps them to know what the user actually likes, what are their needs, expectations as well as their dislikes. The next step of whether a particular visitor could be converted to a customer based on their actions or fulfilled needs is the deciding factor for increasing the revenue or profits of various businesses. There are various dynamic algorithms available which help to predict the intent of the visitor, out of which Naïve Bayes, Random Forest, X G Boost, KNN and Logistic Regression algorithms prove to be the most useful. It was also observed that decision trees proved to be the best choice in providing the desired result. XG Boost was the most efficient algorithm in terms of generating the decision trees and also provided the best F-1 Score. Further improvements could be done in each of the algorithms by tuning more parameters.

References

[1] Sun, Hongfei, Huijuan Wu, Sitong Li, and Min Liu. "The Customer Loyalty Research Based on B2C E-Commerce Sites." In 2010 International Conference on E-Business and E-Government, pp. 3156-3159. IEEE, 2010.

[2] Peker, Serhat, Altan Kocyigit, and P. Erhan Eren. "An empirical comparison of customer behavior modeling approaches for shopping list prediction." In 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1220-1225. IEEE, 2018.

[3] Buckley S, Ettl M, Jain P, Luss R, Petrik M, Ravi RK, Venkatramani C. “Social media and customer behavior analytics for personalized customer engagements”. In IBM Journal of Research and Development, pp. 7-1, 2014 November.

[4] He, Mengxun, Chunying Ren, and Haijun Zhang. "Intent-based recommendation for B2C e-commerce platforms." IBM Journal of Research and Development 58, (2014): 5-1.

[5] Mukhlas, Amalia, Aishah Ahmad, Zahiruddin Zainun, and Media Prima Berhad. "Data mining technique: Towards supporting local co-operative society in customer profiling, market analysis and prototype construction." In 2016 International Conference on Information and Communication Technology (ICICTM), pp. 109-114. IEEE, 2016.