Kojiro So
12/7/22

Predicting Online Shopper Intent: A Comparison of Popular Classification Algorithms

Problem and motivation

E-commerce sales are now an integral part of most modern retail businesses. According to data from the Federal Reserve Bank, e-commerce sales as a proportion of total sales has continued to steadily increase over almost 22 years since 2000. Although temporary, a sharp rise in online sales was also observed during the Covid-19 pandemic. Given this trend, learning the factors that nudge consumers into purchasing goods online is key for any retail business looking to succeed online. There are a plethora of metrics that could be used to try to understand consumer behavior on a website, but only so much can be derived by a human trying to parse this data. Instead, machine learning can be used to predict what factors encourage consumers to purchase (or not purchase) goods during their visit to an online store.

Related work

The original paper that used the dataset that I used in this project is titled “Real-time prediction of online shoppers’ purchasing intention using multilayer perceptron and LSTM recurrent neural networks”. In this paper, the authors explore the use of this same dataset to make real-time predictions rather than static predictions. The idea behind making real-time predictions is that specific strategies like deals and promotions can be presented to a visitor to keep them engaged for a longer time, which results in a higher likelihood of a purchase. I wanted to investigate the usability of a static approach, and further compare the performance of several different models to better understand possible features that consistently affect online shopper behavior.

A paper titled “Classification of E-customer Sessions Based on Support Vector Machine” used an SVM classifier to predict online customer purchases using attributes related to session features such as session duration, number of HTTP requests, and volume of data transfer. They were able to train a model with a predictive accuracy of over 99%, with the probability of predicting a buying session of almost 95%. While these results are promising, a model trained on SVM was too time inefficient for the problem at hand in this project. In addition, the model created was tuned specifically for one particular online store and the authors mention that the model would need to be constructed and tuned for every new online store which is not ideal.

Dataset description

Figure 1: Attributes, Non-Null Count, and Data Type

```
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12330 non-null  int64
 1   Administrative_Duration  12330 non-null  float64
 2   Informational            12330 non-null  int64
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  int64
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object
 11  OperatingSystems         12330 non-null  int64
 12  Browser                  12330 non-null  int64
 13  Region                   12330 non-null  int64
 14  TrafficType              12330 non-null  int64
 15  VisitorType              12330 non-null  object
 16  Weekend                  12330 non-null  bool
 17  Revenue                  12330 non-null  bool
dtypes: bool(2), float64(7), int64(7), object(2)
```

The dataset used for this project is taken from the UC Irvine Machine Learning Repository and consists of 12330 instances, 17 features, and 1 target feature. Six of the features, “Administrative”, “Administrative Duration”, “Informational”, “Informational Duration”, “ProductRelated”, and “Product Related Duration” describe the categories of pages visited by a consumer and the amount of time (in seconds) the consumer spent on pages in each category. “BounceRate”, “ExitRates”, and “PageValues” are numerical metrics measured by Google Analytics that represent, respectively, the rate at which visitors enter the store from a page and then leave from the same page, the rate at which visitors exit the site from a page, and the average value (defined as (revenue+goal value)/number of unique page views) a page generates. “Special Day” is a numerical feature that represents the closeness of a site visit to a “special” day (e.g. Valentine’s Day, Mother’s Day, etc.), ranging from 0 to 1 in 0.2 increments. “Month” is a categorical feature representing the month name of a visit to the site. “OperatingSystems”, “Browser”, “Region”, “TrafficTypes”, and “VisitorType” are categorical features representing the type of visitor to the site, with traffic type describing the type of traffic a customer represents (e.g. organic, direct, or referral), and visitor type describing whether the customer is a new or returning visitor to the website. “Weekend” is a binary feature describing whether the session was during a weekend or not. Finally, “Revenue” is the target feature representing whether or not the visitor completed a purchase.

Approach

I evaluate the performance of popular classification algorithms on predicting whether a visitor completes a purchase based on all available features. The algorithms I investigated include random forest, k-nearest neighbor, Naive Bayes, gradient boosting, and XGBoost.

For data pre-processing, I took a few steps to ensure the data was ready for a model to be trained on. The dataset fortunately had no missing values, but had eight categorical attributes (including the target attribute) that had to be encoded so that they could be used for training. I applied one-hot-encoding to “Month”, “OperatingSystems”, “Browser”, “Region”, “TrafficType”, and “VisitorType”, and converted boolean categorical values to numerical features.

Upon further investigation, I also found that there was a heavy class imbalance in the target feature for a visitor not making a purchase. To ensure this characteristic was carried over to the training process of the model, I used stratified sampling so that the imbalance in the entire dataset was represented by both the training and test datasets. In addition, to address imbalance (which would result in a less rigorous, generalizable model) I also utilized random oversampling, which randomly samples data from the minority class with replacement, and adds them to the training dataset.

With data pre-processing complete, I trained five different models, each using different classification techniques. I evaluated the performance of the models primarily using Receiver Operating Characteristic Area Under the Curve (ROC AUC), but also examined other metrics including accuracy, precision, recall, and F1 score. After training the models using their default hyperparameters, I conducted hyperparameter tuning with randomized search in an attempt to optimize the models.

After training, I examined the features that were considered to be most important (i.e. most impactful on the target feature) and evaluated the performance of all trained models on test data,

Results and Discussion

Table 1: Evaluation of Models on Training Data (No hyperparameter tuning)

	Mean Accuracy	Mean ROC AUC
Random Forest	0.9536	0.9536
Decision Tree	**0.9676**	**0.9996**
K-Nearest Neighbor	0.7921	0.8815
Naive Bayes	0.5687	0.7655
Gradient Boosting	0.8725	0.9458
XGBoost	0.8691	0.9446

Table 2: Evaluation of Models on Training Data (After hyperparameter tuning)

	Mean Accuracy	Mean ROC AUC
Random Forest	0.849	0.923
Decision Tree	0.863	**0.946**
K-Nearest Neighbor	**0.930**	0.934
Naive Bayes	0.626	0.792
Gradient Boosting	0.853	0.921
XGBoost	0.869	**0.946**

Table 3: Evaluation of Models on Test Data

	Accuracy	ROC AUC
Random Forest	0.8504	0.8559

Decision Tree	0.8354	0.8342
K-Nearest Neighbor	0.8341	0.8089
Naive Bayes	0.4084	0.6328
Gradient Boosting	**0.8625**	0.8492
XGBoost	0.8536	**0.8578**

Initial training results as seen in Table 1 show that the model trained using decision trees performed the best when measuring performance by AUC and accuracy, followed closely by random forest, gradient boosting, and XG boost models. The Naive Bayes model performed significantly worse than any of the other models.

Training results after hyperparameter tuning as seen in Table 2 show that both the decision tree and XGBoost models perform identically in terms of AUC, with XGBoost scoring slightly higher in terms of accuracy. K-nearest neighbor scored the highest in accuracy. These results show that a slight performance boost was seen in some models like Naive Bayes and K-nearest neighbor as a result of hyperparameter tuning, while others stayed relatively the same.

The test results in Table 3 show that the model trained using XGBoost performed the best in terms of AUC, while the gradient boosting trained model performed the best in regards to accuracy. The changes in the results from training to test show a significant decrease in both evaluation metrics, indicating that there was slight overfitting on the training dataset, most likely caused by randomized oversampling that was conducted during the data pre-processing steps. Regardless, a model that can predict whether a shopper intends to purchase or not purchase an item with an accuracy of about 86% is a step in the right direction with room for improvement. Deploying the model could allow an online retail business owner to implement strategies to nudge customers towards purchases, especially if these predictions are made in real-time as a customer is shopping and browsing the website.

Figure 2: Feature Importance Derived From Random Forest

```
[(0.654181769698376, 'PageValues'),
 (0.08039120112221466, 'ExitRates'),
 (0.0589872623557751, 'ProductRelated_Duration'),
 (0.053348683101578254, 'Nov'),
 (0.02984397763601644, 'ProductRelated'),
 (0.029061923113899144, 'BounceRates'),
 (0.01632409813609423, 'Administrative'),
 (0.01470290049954839, 'Administrative_Duration'),
 (0.01416210964567522, 'May'),
 (0.007084343970915404, 'Mar'),
 (0.004760690612286044, 'Informational_Duration'),
 (0.0029624463864426024, 'Oct'),
 (0.00283945647493648, 'Informational'),
 (0.002613951529583834, 'Sep'),
 (0.0014162378720287577, 'SpecialDay'),
 (0.0013809228429984106, 'Dec'),
 (0.0009423572867230426, 'Jul'),
 (0.0004657379847505599, 'Aug'),
 (0.000333935805000284, 'Weekend'),
 (9.042919450810386e-05, 'June'),
 (7.538911135128773e-05, 'Feb')]
```

Figure 3: Feature Importance Derived From Gradient Boosting

```
[(0.853117021594681, 'PageValues'),
 (0.05610085090322287, 'Nov'),
 (0.017538104881990986, 'ProductRelated'),
 (0.014020721034137212, 'Administrative_Duration'),
 (0.013179161647129402, 'ExitRates'),
 (0.012799214320433754, 'ProductRelated_Duration'),
 (0.009178389869657482, 'BounceRates'),
 (0.006936682702311715, 'Administrative'),
 (0.006830462933481231, 'May'),
 (0.0016652548810675266, 'Informational_Duration'),
 (0.0006914052274534316, 'Informational'),
 (0.00010981497425896581, 'Weekend'),
 (0.0, 'SpecialDay'),
 (0.0, 'Sep'),
 (0.0, 'Oct'),
 (0.0, 'Mar'),
 (0.0, 'June'),
 (0.0, 'Jul'),
 (0.0, 'Feb'),
 (0.0, 'Dec'),
 (0.0, 'Aug')]
```

Figure 4: Feature Importance Derived From XGBoost

```
[(0.27503315, 'PageValues'),
 (0.08546452, 'Nov'),
 (0.055032693, 'Mar'),
 (0.04631206, 'May'),
 (0.024049016, 'Sep'),
 (0.020678254, 'ProductRelated'),
 (0.018467665, 'BounceRates'),
 (0.016579432, 'ExitRates'),
 (0.016324468, 'ProductRelated_Duration'),
 (0.0155594405, 'Administrative_Duration'),
 (0.015194254, 'Administrative'),
 (0.012255902, 'Oct'),
 (0.012003, 'Dec'),
 (0.010547569, 'Informational'),
 (0.010018959, 'Jul'),
 (0.0089601055, 'Informational_Duration'),
 (0.008354536, 'Feb'),
 (0.00808547, 'Weekend'),
 (0.007955417, 'Aug'),
 (0.006080012, 'June'),
 (0.004736453, 'SpecialDay')]
```

Finally, taking a look at the features considered to be important in predicting whether a shopper would make a purchase or not, “PageValues” was by far the most important feature regardless of which model feature importance was derived from. Considering that page value represents the average value of a page relative to the value that it ultimately generates for a company (whether in terms of real monetary value or not), this is a conclusion that makes sense. Companies should spend resources ensuring that any pages with high page value maintain high page value, and further investigation into the factors that result in a high page value could result in further value gains in other areas of the business. Other features that stood out were the month of November, features related to the duration that a customer spent on specific pages, and the “SpecialDay” feature. Relative to other months, November stood out as being the most important in accurate predictions, indicating that there may be some insights to be gained from further investigating any possible reasons for its importance. Seeing that the duration of some page categories were relatively more important for accurate predictions makes sense intuitively, in that the more time a visitor spends on the website the more time they could be thinking about making purchases. “SpecialDay” stood out as surprisingly low in feature importance, which is notable due to the fact that it is widely understood that some businesses rake in a significant portion of their revenue specifically from holidays and other “special” days like Black Friday.

References

Sakar,C. & Kastro,Yomi. (2018). Online Shoppers Purchasing Intention Dataset. UCI Machine Learning Repository.

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Comput & Applic 31, 6893–6908 (2019). https://doi.org/10.1007/s00521-018-3523-0

Suchacka G, Skolimowska-Kulig M, Potempa A (2015) Classification of e-customer sessions based on support vector machine. ECMS 15:594–600