DATA 1030 Final Project Report

Online Shoppers Purchasing Intention Dataset Analysis

Zuxuan Huai

Brown University

Data Science Initiative

https://github.com/wendyhuai/Online-Shoppers-Analysis

Introduction

Over the last few years, e-commerce has become an indispensable part of the global retail framework. As more people choose to shop online, how and why, when and what makes a customer click on the "purchase" button is the question that many online retailers seek to answer. This project analyzes the online shoppers purchasing intention dataset from UCI Machine Learning Repository and investigates what factors will classify a website visit as a visit with purchase made or without.

This dataset contains 12330 rows and 18 columns. The target variable is the last column of the dataset, which is the revenue column with Boolean values true or false. A true means the user made a purchase in this visit, and a false means the user did not make a purchase. The goal of project is to classify whether a user made a purchase or not given his/her information.

The predictor variables are the first 17 columns (10 numerical and 7 categorical).

- Administrative, Informational, Product Related: the number of different types of pages visited by the visitor in that session
- Administrative Duration, Informational Duration, Product Related Duration: total time spent in each of these page categories
- Bounce Rates: the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session
- Exit Rates: for all pageviews to the page, the percentage that were the last in the session
- Page Values: the average value for a web page that a user visited before completing an e-commerce transaction
- Special Day: numerical [0, 0.2, 0.4, 0.6, 0.8, 1] closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day)
- Month: 10 unique months, excluding Jan and Apr
- Operating Systems: 8 categories
- Browser: 13 categories
- Region: 9 categories
- Traffic Type: 20 categories
- Visitor Type: indicates a user is a returning visitor, new visitor, or other
- Weekend: (Ture/False) whether the visit date is a weekend

A real-time online shopper behavior analysis system has been built using this dataset. The researchers have successfully predicted the purchasing intention using clickstream and session information data.

EDA

Target Variable Analysis

The target variable has 10422 false values and 1908 true values. Figure 1 visualizes the breakdown of the target variable.

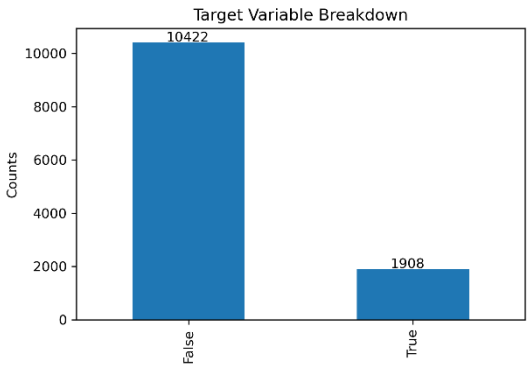

Figure 1: Target Variable Breakdown

Predictor Variables - Numerical Variables

The first 6 columns include information about number of visits in a certain type of page and the total time spent on this type of page. The types include administrative, informational, and product related. It is worth noticing that the number of product-related page visits shows a

positive correlation with the time spent on product-related pages. Figure 2 shows the scatter plot of these two variables.

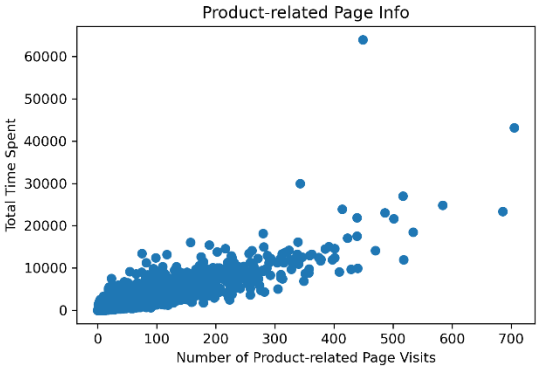

Figure 2: Product-related Scatter Plot

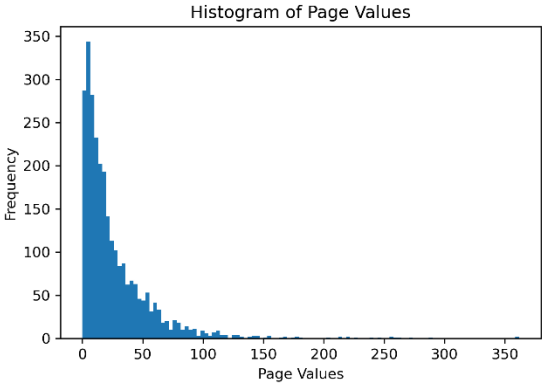

Figure 3: Histogram of Page Values

The next 4 columns are numerical variables as well. After looking at the brief statistics of these columns below, I noticed that the majority of Page Values and Special Days are zero. Since 77.9% of Page Values are zero, a histogram excluding the zeros gives an overview of the distribution of the remaining data, shown in Figure 3.

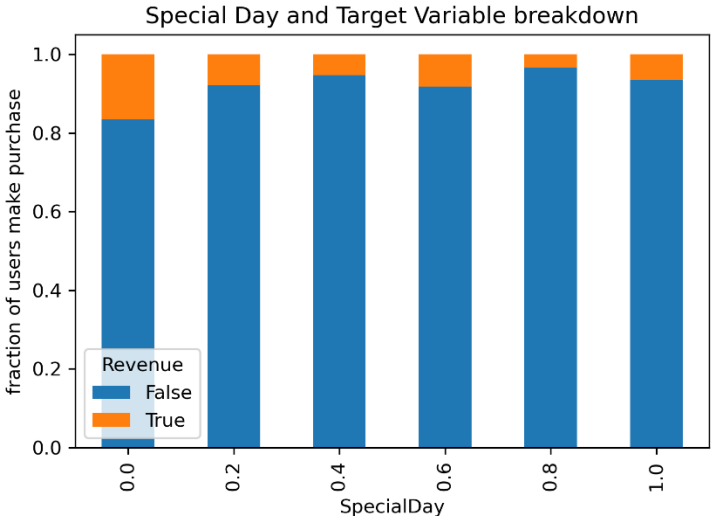

Figure 4: Special Day and Target Variable Breakdown

Since The special day measures the closeness of the purchase date to a special date, I will treat this variable as ordinal variable in future analysis. Figure 4 shows the fraction of users make purchases with the corresponding special date variable. It is interesting to see that fewer people make purchase at 0.4 and 0.8 compare to other special day numbers.

Predictor Variables - Categorical Variables

There are many categorical variables in this dataset, here is the breakdown of month and purchases.

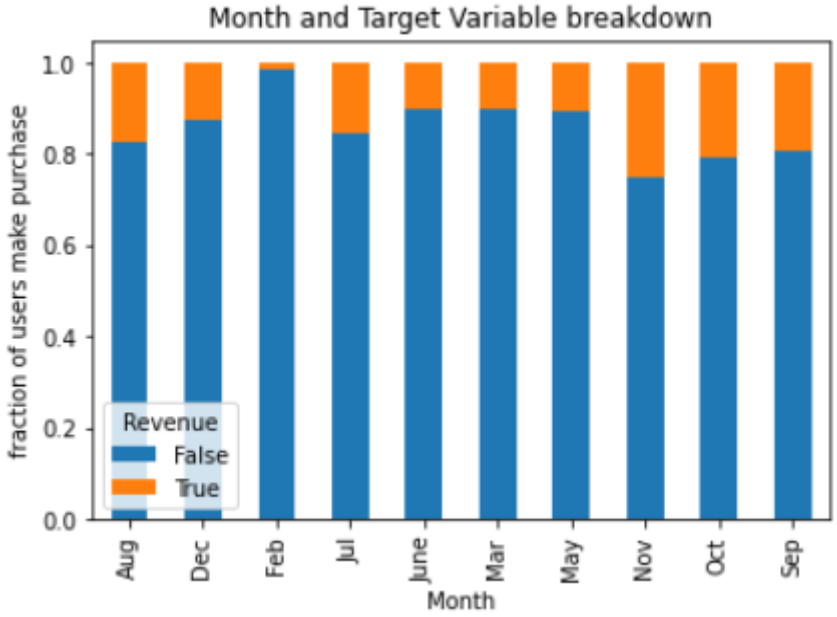

Figure 5: Month and Target Variable Breakdown

This figure shows the fraction of purchases make in each month. January and April is not present in the dataset. A small fraction of users makes purchases in Feb and a relatively large proportion of users make purchases in their visits in November.

Methods

Splitting Strategy

Since each rows represents a unique visit from an anonymous users, this dataset is iid. Hence, it does not have a group structure, and it is not a time series data. Because only a small portion of the dataset has true in the revenue column, I will perform a Stratified K-Fold data split to ensure there are similar fraction of trues in target variable in the training, validation, and testing dataset. I will use the standard 60%, 20%, 20% data split.

Data Preprocessing

From exploratory analysis, I find many numerical variables are heavily tailed or has a few extremely large values, so I will use Standard Scaler to transform the continuous variables. The Special Date variable although it is numerical, it represents the closeness of the purchase date to the special date, I will treat it as a categorical feature and use Ordinal Encoder to transform. I will transform the remaining categorical features using OneHotEncoder. The target variables are true and false, which does not need any transformation.

There are 17 features in the processed dataset. After preprocessing, the 17 predictor columns have expanded to 74 columns.

Machine learning Pipeline

In order to train various classification models, I constructed a function called MLpipe_KFold_Accuracy to process the data and fit models. The inputs of this function include the feature dataset X, the target variable y, the preprocessor which preprocess the data, the machine learning algorithm to train, and the parameter grid for hyperparameter tuning. This function will loop through five random states to include the uncertainties introduced by training data variation through stratified cross validation. Under each random state, the data is first randomly split into training set and test set. Next, the training set is used in a four-split stratified cross validation. In each fold, the data go through the pipeline that first preprocess the data and then fit the input machine learning algorithm. Various combinations of hyperparameters are tested to determine which parameters give the best validation score. The selected model is used to fit the test set and the accuracy score is calculated. Lastly, the function returns the best models and test scores from the five random states.

I use accuracy score as the measurement for model performance because it is the most straight forward way. I will dig deeper into precision score, recall score, and f-score in the best performing model.

Training Models

- **Logistic Regressions**
  In logistic regressions, I trained models with different penalty parameter separately. I set the solver as saga and the max iteration is 2000. When penalty is 'l1', the model is lasso regression. When penalty is 'ls', the model is ridge regression. When penalty is 'none', the model is logistic regression without any penalties. Under each type, I tuned the parameter C between 1/log(-2) and 1/log(2). For elastic net, in addition to parameter C, I tuned the parameter l1 ratio between 0 and 1.

  In each iteration, the best parameter varies. This is no single value that stand out among the models, the hyperparameter really depends on the training data.

- **Random Forest**
  In random forest, I tuned the max depth and max features parameters. I tried 1, 3, 10, 30, and 100 for max depth and 0.5, 0.75, and 1.0 for max features. In four out of five models, the model performs the best when the max depth equals to 10, and all the models performs the best when max features equals to 0.5.

- **SVC**
  In support vector classifier, I tuned two parameters: C and gamma. I tuned C in range 1e-03 to 1e04, and gamma in the same range. Three out of five models performs the best when C equals to 1.0 and gamma equals to 0.1, and the other two performs the best when gamma equals to 0.01. The parameter tuning depends on the input data as well.

- **K-Nearest Neighbors**
  For k-nearest neighbors classifier, I tuned three parameters: number of neighbors, leaf size, and weight methods. For number of neighbors, I used 5, 10, 15, 20, and 25. For leaf size, I used 10, 30, and 50. I tried two weight methods: uniform and distance. The results are pretty uniform, all five models perform the best when leaf size is 10 and the weight method is uniform. The number of neighbors depend on the training dataset. Two models prefer 15 neighbors, two models prefer 10 neighbors, and one prefers 25 neighbors.

Results

The table below summaries the accuracy scores from various machine learning algorithms. The baseline model has the accuracy score of 0.8453. All models have improved the accuracy scores. Random Forest has the best average test accuracy score of 0.9027.

Model	Average Accuracy Score
Baseline Model	0.8453
Logistic Regression	0.8839
Lasso Regression	0.8835
Ridge Regression	0.8837
Elastic Net	0.8833
Random Forest	0.9033
SVC	0.8921
K-Nearest Neighbors	0.8747

Table 1: Model Accuracy Summary

The standard deviation of random forest accuracy score is 0.00596, which makes the prediction score 9.74 standard deviations above the baseline model.

Random Forest

The table below shows a more detailed breakdown of the model performance.

Accuracy Score	0.9033
Precision Score	0.7300
Recall Score	0.5969
F Score (Beta = 0.5)	0.6987
F Score (Beta = 1)	0.6566
F Score (Beta = 2)	0.6194

Table 2: Random Forest Scores

Out of all predicted users visits who will make a purchase, 72.54% are actually made a purchase. Out of all user visits with a purchase made, 60.26% are correctly predicted. From the higher precision score, we can tell that the model produces more false positives than false negatives, indicating there are more user visits without a purchase made are predicted as with purchase than visits with purchases made but are predicted as a normal visit.

Feature Importance

I analyzed feature importance using permutation feature importance and feature importance attribute from random forest

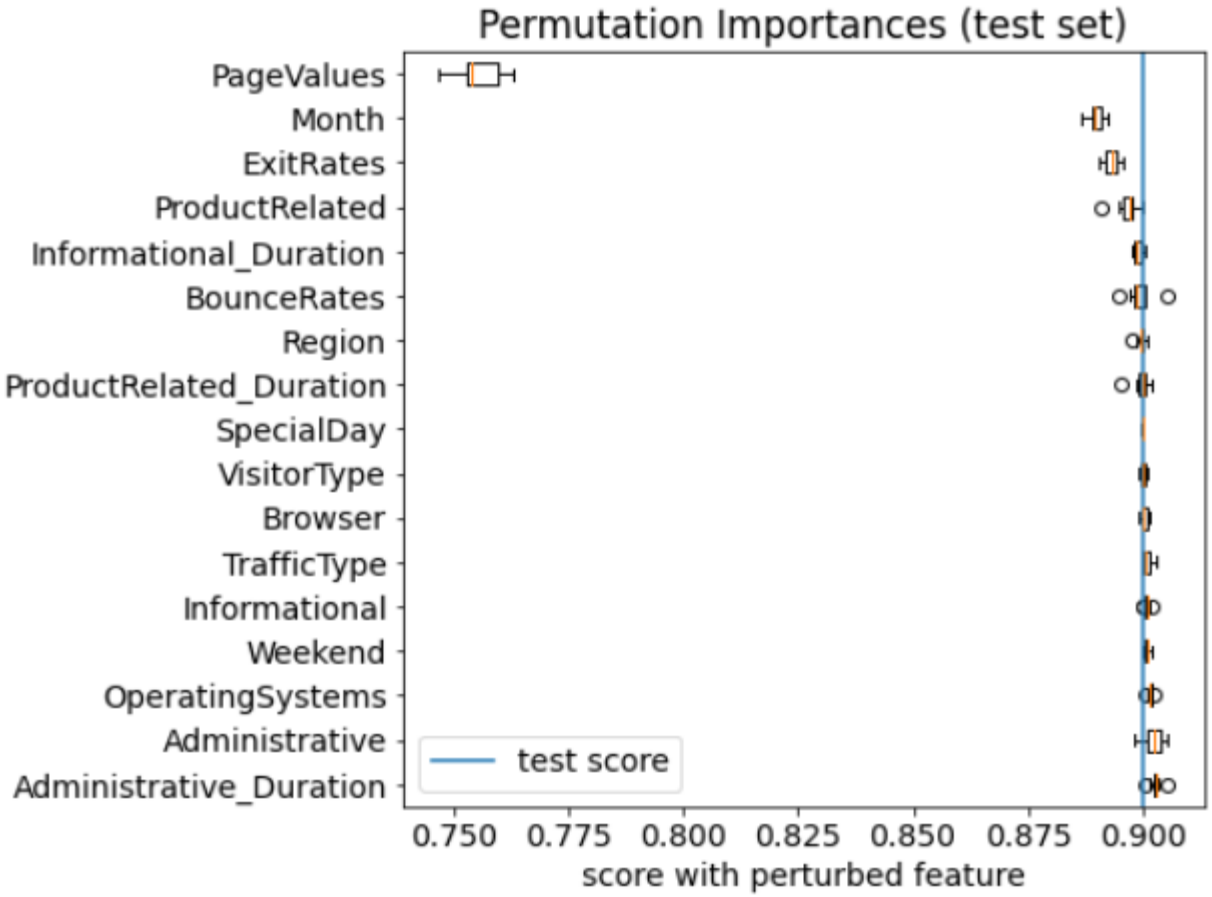

Figure 6: Permutation Importance

From permutation test, the three most important features are PageValues, Month, and ExitRates. From random forest, the top three most important features are PageValues, ExitRates, and Product Related Duration. Page values are the average value for a web page that a user visited before completing an e-commerce transaction. Product Related Duration measures the total time spent a user spend on product related webpages. It makes sense because users tend to browse many products before completing a purchase. It is surprising that Month is the second most important variable, whereas Weekend is a not very important variable.

These features provide guidance for the company's marketing strategy to increase number of visits with a purchase. The company can polish product related pages to increase the purchase rate. Because month is also an importance factor, the company can increase promotion in certain months to attract more customers.

Outlook

Although the accuracy score seems to be high, due to the high baseline score, the precision and recall scores still have room for improvement. A model I did not cover in this project is the XGBoost, which is a more advanced tree-based model. I also did not come up with any new features. I could do some feature engineering which might boost the accuracy rate. Training some

of these models take a long time. If we could find a more efficient way for building models, online stores can develop a real-time online shopper behavior analysis system to monitor user activities.

Reference

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018).