DATA 1030 FINAL PROJECT REPORT

Online Shopper's Intention Classification and Prediction

Shiqi Lei

Brown University

Data Science Initiative

Github Repository: https://github.com/beckyleii/data1030-f19-project.git

I. Introduction

The project utilized the data set from Kaggle – Online shopper's intention. The data was obtained from the URL information and Google Analytics for each pages in the e-commerce site. One of the reason why this dataset is meaningful to work on is that the e-commerce becomes more and more popular nowadays, and it has already embedded in our daily life especially for college students and people who are too busy to go shopping in stores. Thus, the dataset has great potential in analyzing the trend of decision making by visitors in making purchases, and the result might suggest what improvement the e-commerce site might do to stimulate the revenue. In this project, we will try build a revenue predictor for one such website, to be more specific, I will predict whether the revenue will be generated based on the duration and special period of the time, and I will also analyze which feature vector is important among e-commerce categories while having huge impact on the target variable 'Revenue'.

As for the dataset, it consists feature vectors belongs to 12,330 sessions, with classification target variable "Revenue" and "was formed so that each session would belong to a different users in a 1-year period to avoid any tendency to a specific campaign, special day, user profile or period" (Sakar, 2019). The dataset consist of 10 numerical and 8 categorical attributes including the target variable. In this dataset, we are setting 'Revenue' attributes as the target variable, which returns Boolean value indicating whether the user generates revenue or not. First we have three features: "Administrative", "Informational", "Product Related", which are three different types of pages that were visited by the visitors in that session, and respectively, there are three numerical data: 'Administrative Duration', 'Informational Duration', 'Product Related Duration', which represent the total time visitors spent in each of these page categories. The

values are derived from the URL information and the information are updated when a user moving from one page to another (Sakar, 2019).

Then there are three categorical data: ‘Visitor Type’, which returns categories of ‘returning visitor’, ‘new visitor’ or ‘others’; ‘Weekend’, which returns Boolean value indicating whether the date of the visit is weekend; ‘Month’, which keeps track of the month of the year that the session occurs, as well as ‘Traffic Type’, ‘Region’, ‘Operating Systems’, which use numerical values to represent different categories. The dataset also have categorical feature ‘Browser’, indicating the type of browser that the visitor used; ‘Operating System’, ‘Region’ and ‘Traffic Type’ of the session. Then we have numerical features: 'Bounce Rate', which refers to the percentage of visitors who enter the site from that categorical page and then leave (‘bounce’) without triggering another request to the analytics server during that session; ‘Exit Rate’, referring to the percentage that the specific web page were last in the session; ‘Special Day’, measures the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction, and lastly ‘Page Value’, which represents the average value for a web page that a user visited before completing an e-commerce transaction (Sakar, 2019).

Moreover, from the Kaggle public project, one of the project utilized the clustering analysis to learn the user characteristics in terms of time spent on the websites. And another public project uses the logistic regressing classification, however, the project did not have any information about how the data were sued and what result was derived from the data.

II. EDA

As for the exploratory data analysis, we firstly check the balance of the dataset, obtained 84% for the False class in 'Revenue', and 15% for the True class in our target variable. A 15% target incidence considering the amount of data in this dataset is reasonable for the real life dataset, thus we do not need to implement data augmentation techniques for the imbalance. There are two feature vectors that behave differently from others during the analysis. The first one is the 'Page Value'. As we can see in the Figure 1 above, the strip plot of 'Bounce Rate vs. Revenue' and 'Exit Rates vs. Revenue' behaves similarly as more users tend to withdraw the page if they do not make the purchase. And in the 'Page Value vs. Revenue', users

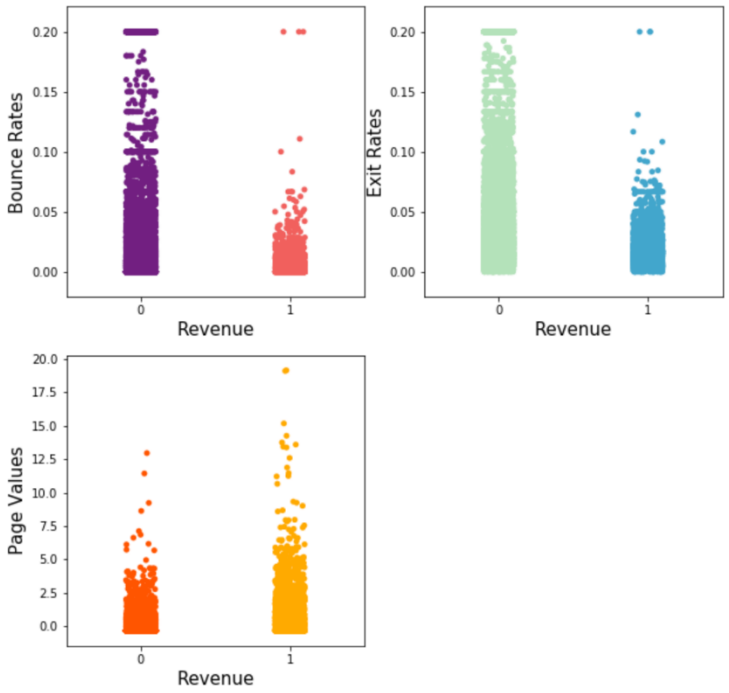

Figure 1. Strip Plot (Bounce Rate vs. Revenue; Exit Rate vs. Revenue; Page Value vs. Revenue)

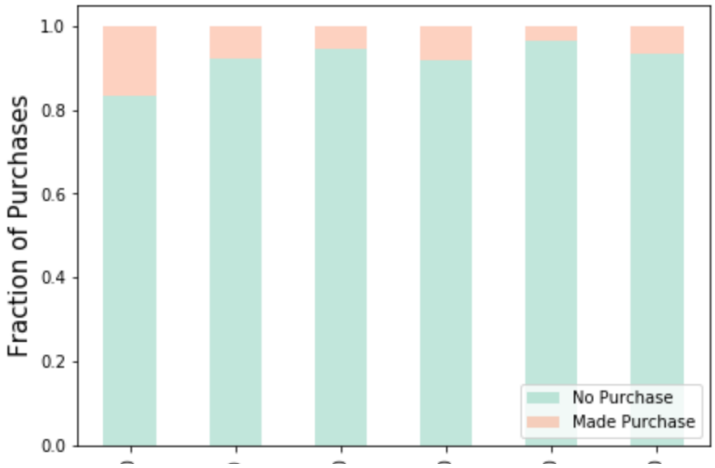

Figure 2. Stack Bar Plot (Special day vs. Revenue)

who make purchases tend to visit more pages. Also different from what we expected, the closeness of ‘Special day’ does not have too many impact on the action of purchase (see Figure 2 above). This behavior is also demonstrated in the correlation matrix (see Figure 3 below).

The correlation matrix in Figure 3 shows how each features is related to other feature vectors including target variables, and we can see that the ‘Page Value’, ‘Product Related’ are top two positively correlated features with respect to ‘Revenue’, ‘Exist Rates’ and ‘Bounce Rates’ are top two negatively correlated features to target variable. Not surprisingly, the Special Day has a correlation of -0.082468 to the ‘Revenue’, meaning that it only has small impact on target variable.

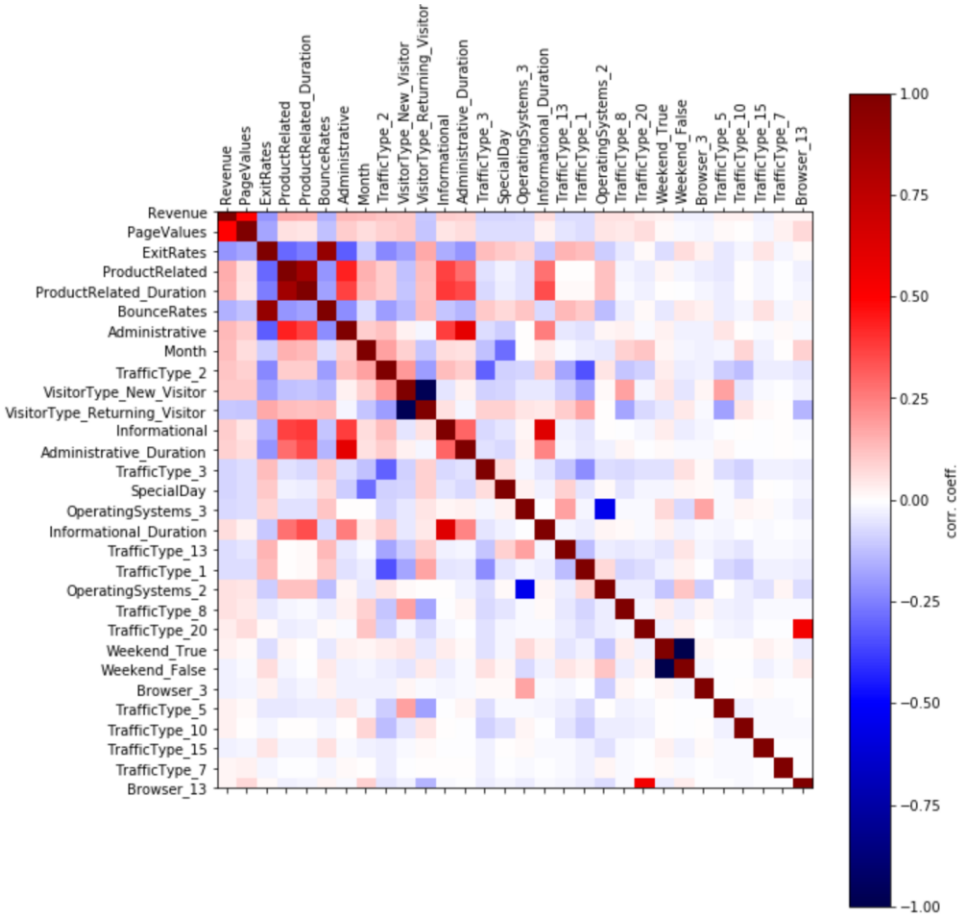

Figure 3. Correlated Matrix of Preprocessed Features

III. Methods

Preprocess

In handling missing data points, I found that there are 14 missing data points. Since we only have 14 points out of 12,330 data points missing, and we obtain MCAR test with p-value of 0.23, we conclude that the missing data is missing at random, thus we can delete the entire row with the missing data points.

As I went through the dataset, it seems that three durations are tailed distributed, and thus, we apply Standard Scalar on the continuous features: ‘Administrative Duration’, ‘Informational Duration’, ‘Product Related Duration’, ‘Administrative’, ‘Informational’, ‘Product Related’, and ‘Page Value’. In addition, ‘Browser’, ‘Operating System’, ‘Traffic type’, ‘Visitor type’ , ‘Weekend’, and "Region" are unordered categorical data, with "Weekend" as binary categorical data, all of these can use One Hot Encoder. Then we have ‘Month’ and ‘Special Day’ both as ordered categorical data, hence we can use Ordinal Encoder to fit. It is important to note that the percentages such as ‘Bounce Rates’ and ‘Exit Rates’ are already between 0 and 1, and therefore we do not need to process these two features. At last, we processed the target variable ‘Revenue’ using Label Encoder since it is a binary categorical data that returns True and False. After encoding, initial 18 input features increased to 67.

Cross Validation and Machine Learning Pipeline

As for the machine learning pipeline, I used GridSearchCV approach to refits the best model to X_other and y_other, which both sets were derived in the previous step when I split my independent and identically distributed data into other_data and test_data, for k-fold cross validation with 20% test set and 80% other set. And I utilize stratified K-folds cross-validators to

preserve the percentage of samples for each class in order to tackle the imbalance issue. Before calculating the accuracy score of models, I used preprocessor pipeline to preprocess 17 features. Combining preprocessor and supervised machine learning method, I calculate the test accuracy score as the result. For each model, I calculate the test score for 10 different random state with 4 folds, and produce the mean and standard deviation test score from random test sets in the end. Each time when a random state is fixed, the K-fold GridSearchCV will iterate through all the combinations of the hyperparameters we are trying to tune, and produce the best parameter and best test score for this random state. At this stage, we also need to avoid models picking the edge of parameter values range. This project adopted four supervised machine learning approaches: SVC, Random Forest, Logistic Regression as well as K-nearest Neighbors Classifier. After the modeling, accuracy scores are used to evaluate the performance of these approaches due to the fact that the data is not extremely imbalanced comparing to other real-life data. The Table 1 shows the parameters and values I tuned in each model.

Table 1. Parameters and Its Range Values for Models

	Parameter and Range Values for Best Fit
SVC	C : 1.e-01, 1.e+00 1.e+01 1.e+02 1.e+03 Gamma: 1.e-02 1.e-01 1.e+00 1.e+01 1.e+02
Random Forest	Max Depth: 8, 11, 30, 50, 100 Min Sample Split: 8, 16, 32, 64, 128
Logistic Regression	Penalty: L1, L2 C: np.logspace (0, 4, num = 10)
K-nearest Neighbor	Weights: Uniform, Distance N – Neighbors: 10, 20, 30, 40, 50, 60

IV. Results

The accuracy test score for all models are summarized in Table 2 (See below). And we can see that the support vector classifier has highest accuracy scores in average and Random Forest also have great performance similar to SVC. With our baseline model score of 84.49%, SVC and Random Forest has improved 5%.

Table 2. Test Scores for SVC, Random Forest, Logistic Regression, K-nearest Neighbors

	Baseline	SVC	Random Forest	Logistic Regression	K-nearest neighbors
Accuracy Score Mean ± Std	0.844967	0.8932 +/- 0.0014	0.8931 +/- 0.0014	0.8815 +/- 0.0015	0.877 +/- 0.0014

After we find the best fit model for our dataset, I performed permutation feature importance as our global feature importance, which can be understood as a model inspection technique. It is defined to be the "decrease in a model score when a single feature value is randomly shuffled" (Breiman, 2001). Thus, from the figure 4 below, we can conclude that 'Page Value' is an important features, since if it is shuffled, the model score will decrease from around 0.89 to 0.76.

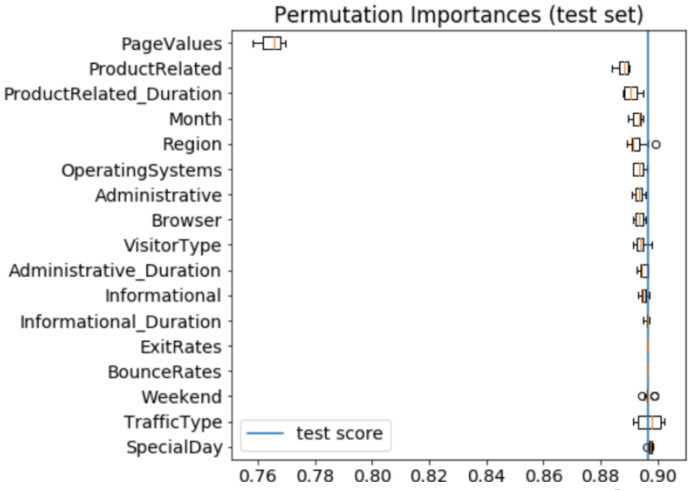

Figure 4. Permutation Importance Plot

At last, we can evaluate the effectiveness of our model using confusion matrix. The matrix shown in Figure 5 is average probability of 10 different random state runs for SVC model. We can see that the probability of correct prediction out of all the positive classes is only 47%, meaning that the model is not very good at predicting whether the users will make the purchase or not based on given features and datasets.

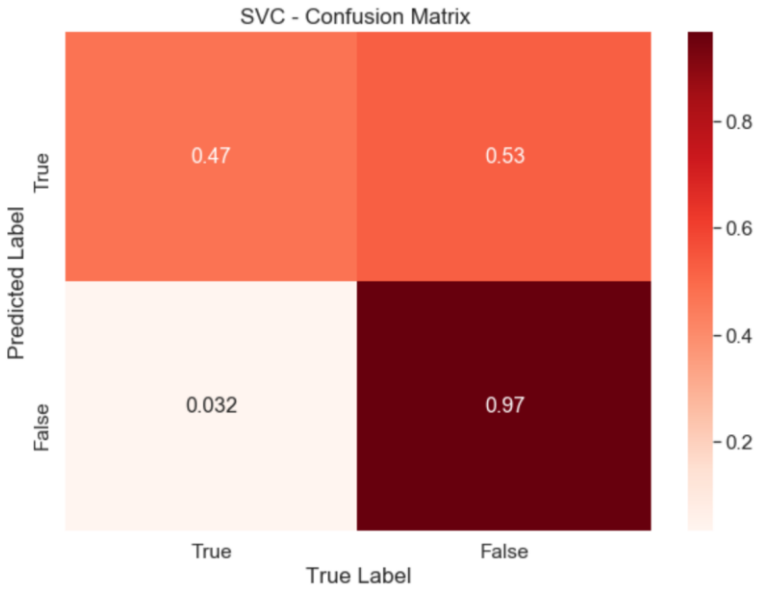

Figure 5. SVC'S Confusion Matrix

Even though the model is not very accurate in predicted positive class correctly, it works great in predicting negative class correctly, meaning that we predicted that the users did not make purchases and he or she actually is not. We can utilize this results and figure out features affecting no-purchasing behavior and, in the future, we might be able to use the analysis to make suggestions on the feature improvement of the commercial website. Perhaps, if we successful improve our model and achieve highly accurate prediction on revenue predictor, knowing the behavior whether the person will make the purchase or not will holds a massive business value for this commercial website.

V. Outlook

Based on confusion matrix, and our scale of improvement after fitting the data, the interpretability of the model still need to be improved, and the reason might because we have a relatively imbalanced dataset, as the result, our model perform weakly in predicting the purchases. The issue might be tackled using Synthetic Minority Over-sampling Technique (SMOTE), which is a over-sampling technique to create synthetic data for categorical as well as quantitative features in the data set (Dissanayake, 2019). According to the previous work with SMOTE technique, the accuracy score can be improved from 84% to 95% after applying it. In addition, since the accuracy score of random forest and SVC are very similar, we might also apply confusion matrix for the random forest to check if it produce better confusion matrix. Moreover, the model performance might also be improved by either obtain more data points for the positive class (‘Revenue’ = True), or include additional features such as the age or sex of the users.

References:

Sakar, Okan C., and Yomi Kastro. “Online Shopper's Intention.” Kaggle, 23 May 2019, https://www.kaggle.com/roshansharma/online-shoppers-intention.

Dissanayake, Isuru. “Online Shoppers Purchasing Intention: Randomforest: ML.” Medium, Analytics Vidhya, 7 Oct. 2019, https://medium.com/analytics-vidhya/ospi-mul-randomforests-156acdb73fd9.

Breiman, Leo. “Random Forests”, Machine Learning, 45(1), 5-32, 2001.