DSC478 Report

Online shopping intention

Introduction

Our findings support the feasibility of accurate and scalable purchasing intention prediction for virtual shopping environment using clickstream and session information data.

The increase in e-commerce usage over the past few years has created potential in the market, but the fact that the conversion rates have not increased at the same rate leads to the need for solutions that present customized promotions to the online shoppers. In physical retailing, salesperson can offer a range of customized alternatives to shoppers based on the experience he or she has gained over time. In this paper, we are aimed to explore customer behavioral and market tendency based on a virtual shopping environment.

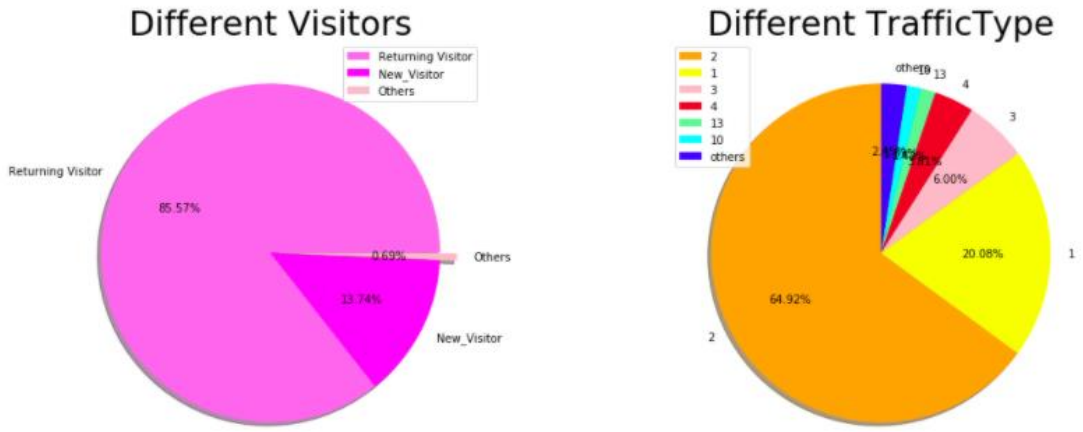
In our study, the purchasing intention model is designed as a binary classification problem measuring the user's intention to finalize the transaction. And it includes 18 variables and 10 of them are numeric variables, the other 8 variables are categorical variable. In this paper, we can simultaneously predict visitor's purchasing intention and likelihood to abandon the site. Thus, we aim to offer content or sales only to those who intend to purchase and not to offer content to the other users. The dataset consists of feature vectors belonging to 12,330 sessions. The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period. Of the 12,330 sessions in the dataset, 84.5% (10,422) were negative class samples that did not end with shopping, and the rest (1908) were positive class samples ending with shopping.

In this case, we are able to create a more accurate model of online shopper intention and provide a clear visualization for salesman under different conditions.

By analyzing this model, we apply several algorithms to tackle this binary classification problem: K nearest neighbors, decision tree, Naïve Bayes tree and random forest.

Visualization

Visualization is a visual method to prove our assumption and sometimes provide a different perspective. For example (*Visualization-fig1:Pie Plot*), “TrafficType” describes the source by which the visitor has arrived at the Web site (e.g., banner, SMS, direct), and “VisitorType” labeled customers as “New Visitor”, “Returning Visitor”, and “Other”. For these two pie plots, we can conclude that most revenue are depending on returned visitors, it takes 85.57% of the total visitors. As for Traffic type, type-2 takes 64.92% in total, and next type-1 takes 20.08%. Thus, we can generalize / promote more information about our webpage or product through type-2 or type-1. These two types might be the most convenient way to receive messages for our customers.



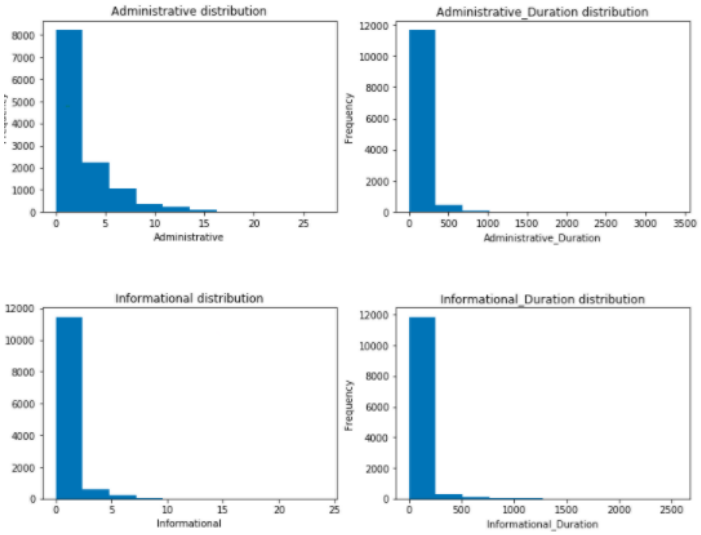
Visualization-fig1: Pie Plot

For the basic information (*Visualization-fig2: intention_df info*) of our dataset, we find that some of these variables only have min and max (such as: informational, Pagevales), which means our dataset has a sparse data. Besides, it shows most visitors are looking through the webpages around 80 minutes on average, and normally will check multiple related products to compare. Thus, we should modify our recommend system, to provide a better product related services for our customers, which might increase their willing to buy a stuff.

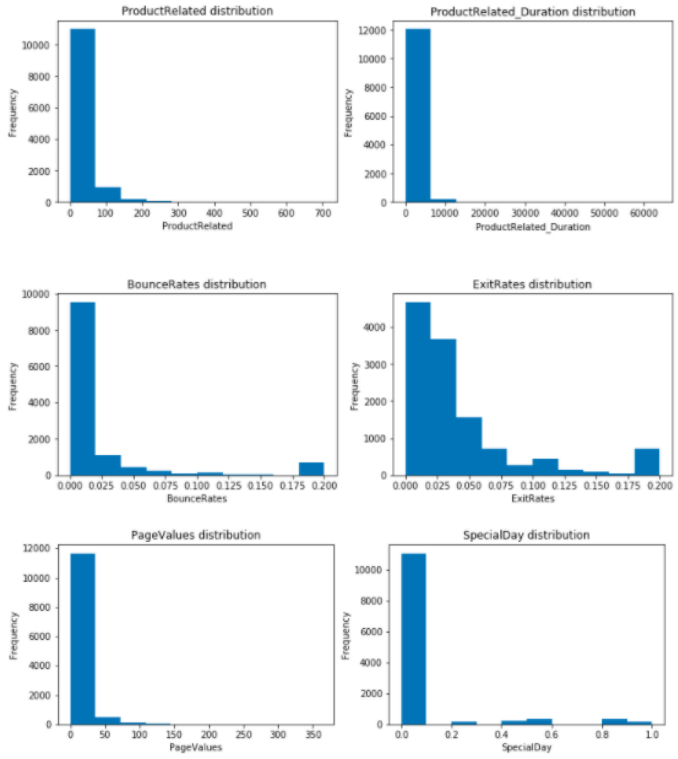
	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues
count	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	2.315166	80.818611	0.503569	34.472398	31.731468	1194.746220	0.022191	0.043073	5.889258
std	3.321784	176.779107	1.270156	140.749294	44.475503	1913.669288	0.048488	0.048597	18.568437
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000	184.137500	0.000000	0.014286	0.000000
50%	1.000000	7.500000	0.000000	0.000000	18.000000	598.936905	0.003112	0.025156	0.000000
75%	4.000000	93.256250	0.000000	0.000000	38.000000	1464.157213	0.016813	0.050000	0.000000
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230	0.200000	0.200000	361.763742

Visualization-fig2: intention_df info.

As we can see from Histogram (*Visualization-fig3&4: Histogram (Numeric Variables Distribution)*), all these models are right skewed, which illustrates their mean is larger than the mode and median. So, we can infer that most customers are just look through the webpages and choose to leave. Only a few of them finished the trade, as we can see, the BounceRates mainly concentrates on (0-0.05), but ExitRates has much more expanded range compared with BounceRates.

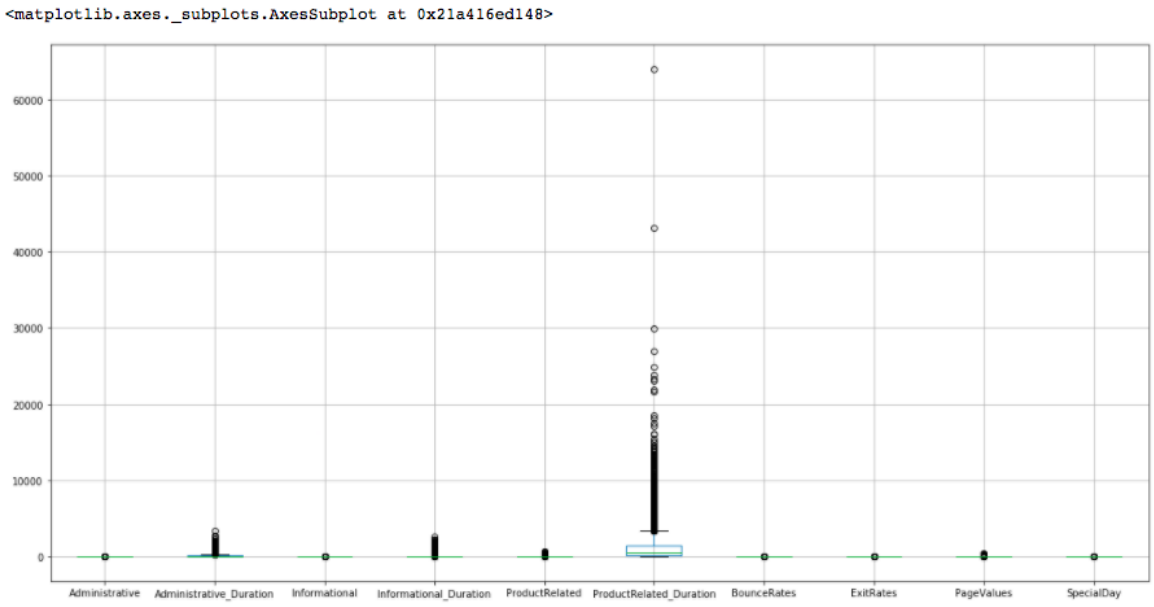


Visualization-fig3: Histogram (Numeric Variable Distribution)



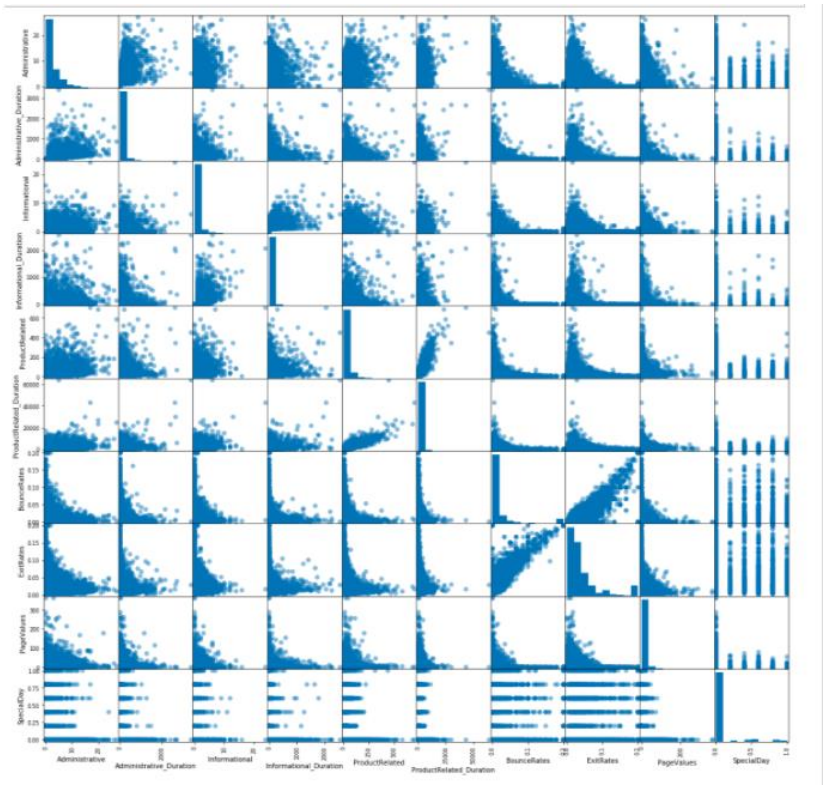
Visualization-fig4: Histogram (Numeric Variable Distribution)

For boxplots (*Visualization-fig5: Boxplots (for Numeric Variables)*), we found an interesting variable, which is ProductRelated_Duration, it has tons of outliers and cost the longest time among all the variables. As our expects, customers usually take a lot of time to compare two or more items at the same time, that's might the reason why ProductRelated_Duration behave like this.

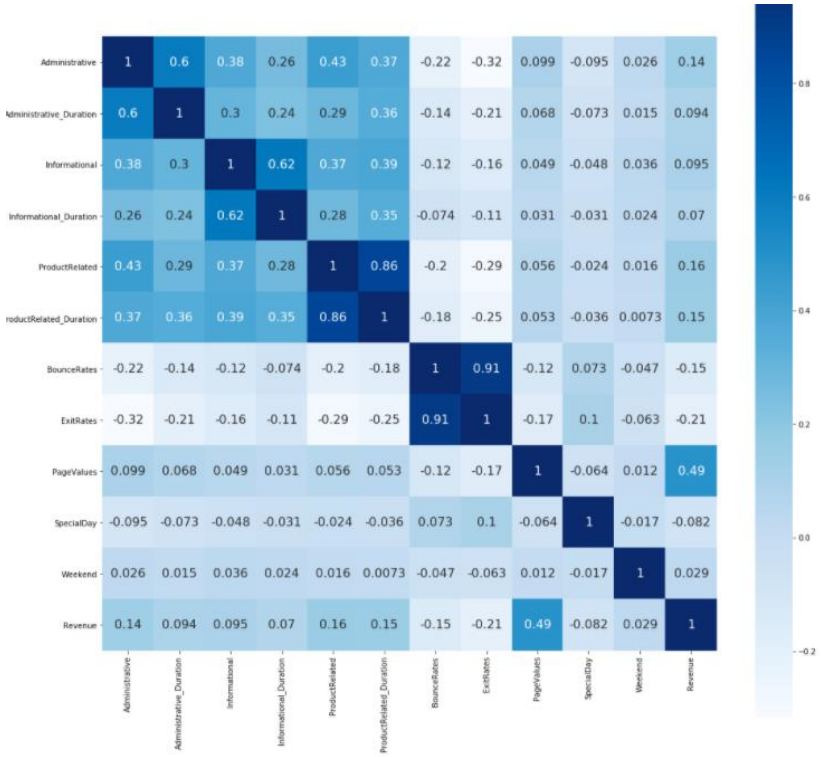


Visualization-fig5: Boxplots (for Numeric Variables)

Correlation matrix is a great method to evaluate relationship among variables. Based on the matrix (*Visualization-fig6: Correlation Matrix*), the Page Value and Administrative, ExitRates and BounceRates are highly correlated, which means with the increase of EixtRates, the BounceRates also increased. Thus, we decided only to keep ExitRates. And for more vivid visualizaion, we also create a correlation matrix plot (*Visualization-fig7: Correlation Matrix (with plots)*). As we can see, variables have lower correlation / inter-correlated between each other.

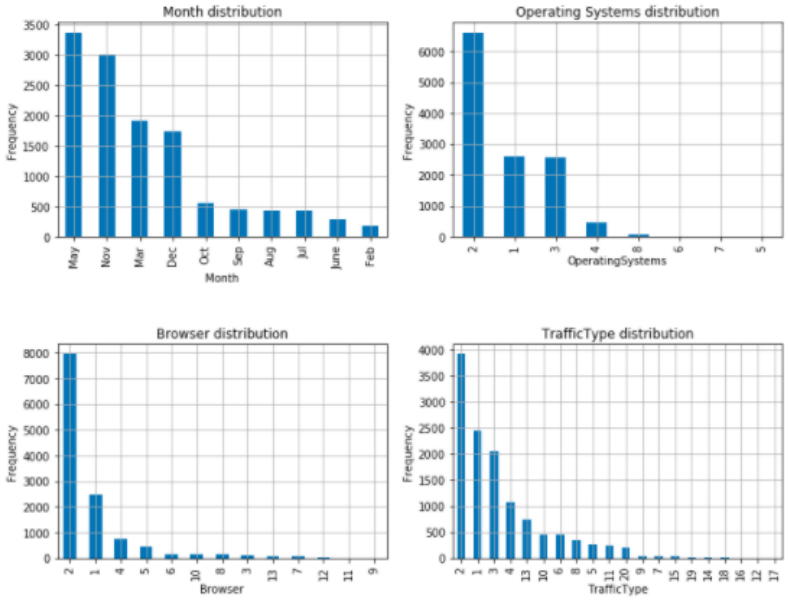


Visualization-fig6: Correlation Matrix (with plots)

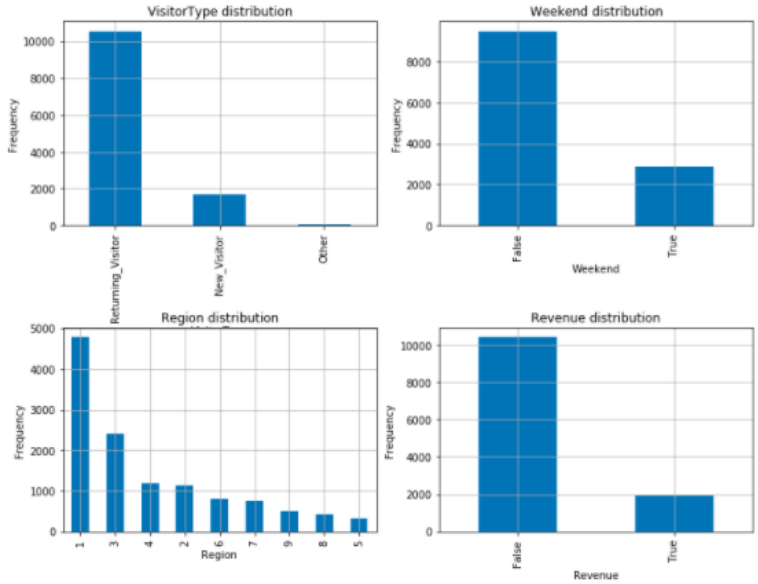


Visualization-fig7: Correlation Matrix (with numbers)

'Month', 'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType', 'Weekend', 'Revenue' are eight categorical variables in our dataset, and “Revenue” is the dependent variable. According to the histogram (*Visualization-fig8&9: Histogram (Categorical Variables)*), which means most potential buyers mainly concentrate on only fewer subjects. Such as, Operating System, Browser and Traffic Type. Thus, one way to increase the revenue is recommend our product through Traffic Type-2, Operating System-2 and Browser Type-2, which could represent the most convenient way for customers to look for products.



Visualization-fig8: Histogram (Categorical Variables)



Visualization-fig9: Histogram (Categorical Variables)

From all these visualizations, we have a better understanding of our dataset, the frequency of each variable, the correlation between variables and different perspective to visualize our model.

Classification

Before implementing classification, we remove highly correlated attributes (ExitRates, ProductRelated_Duration), similar content attribute (SpecialDay) and unrelated attributes (Browser, Region, TrafficType). After that, the dataset is converted to standard spreadsheet format. Then we split the dataset (all predictors and target) into 80/20.

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(9864, 21)
(2466, 21)
(9864, 1)
(2466, 1)
```

Fig-1: Training / Test Shape

1. KNN Classifier

In order to determine the fitted K parameter, we run the classifier with K from 5 to 100 with step of 5 and plot the corresponding training dataset and testing dataset. Through the analysis, the accuracy of training dataset remains the same, but testing dataset has a maximum at K=10, then decreases after that. Therefore, we choose K=10 in our KNN classifier.

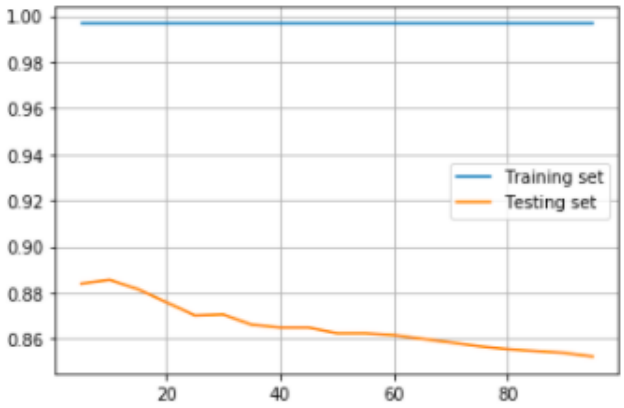


Fig-2: Accuracy of Train/Test among different K values

```
KNN classification report is:
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      2081
           1       0.74      0.41      0.53       385

    accuracy                           0.89      2466
   macro avg       0.82      0.69      0.73      2466
weighted avg       0.87      0.89      0.87      2466

Confusion matrix:
 [[2026   55]
 [ 227  158]]
Train score is 0.9965531224655312, and test score is0.8856447688564477.
```

Fig-3: KNN Classification Report

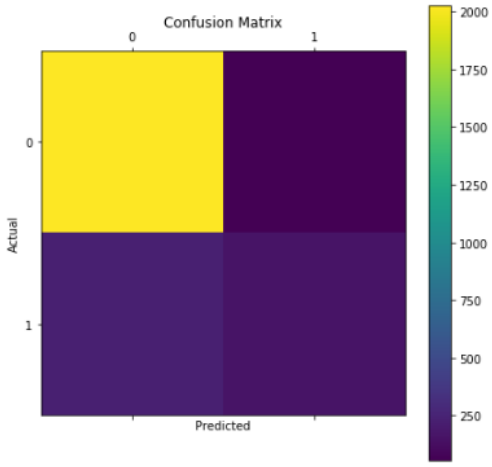


Fig-4: KNN Confusion Matrix:

2. Decision Tree Classifier

We test different parameters for decision tree classifier, including the criterion, min_samples_split and max_depth on training – testing dataset and cross-validation. For the criterion, gini behaves better than entropy on testing dataset.

```
Accuracy for training data:  0.9965531224655312
Accuracy for test data:  0.856853203568532
------------------------------------------------------------------
[0.88848337 0.89537713 0.84630981 0.81914031 0.81549067]
Overall Accuracy: 0.85 (+/- 0.07)
```

Fig-5: Accuracy of Decision Tree Classifier (criterion=entropy)

```
Accuracy for training data:  0.9965531224655312
Accuracy for test data:  0.862124898621249
------------------------------------------------------------------
[0.89781022 0.89618816 0.83333333 0.81103001 0.82562855]
Overall Accuracy: 0.85 (+/- 0.07)
```

Fig-6: Accuracy of Decision Tree Classifier (criterion=gini)

For the min_samples_split, we test a range from 3 to 20 with a step of 1 and choose min_samples_split=18.

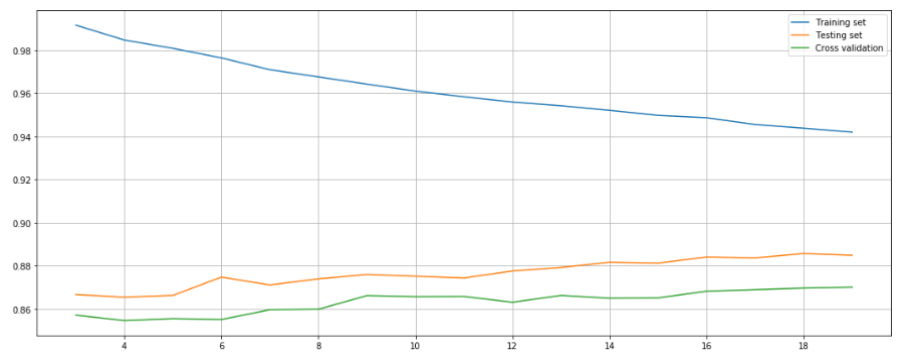


Fig-7: Accuracy of Decision Tree Classifier (criterion=gini)

For the max_depth, we test a range from 5 to 10 with a step of 1 and choose max_depth=5.

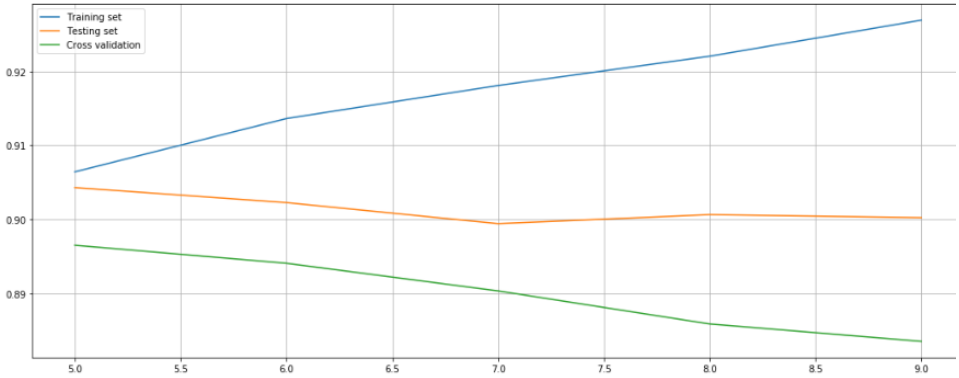


Fig-8: Accuracy of Decision Tree Classifier (criterion=gini, min_samples_split=18)

After that, we build the final model using the above parametersL

criterion = gini
min_samples_split =18
max_depth = 5

```
Accuracy for training data:  0.9064274128142741
Accuracy for test data:  0.9042984590429846
Classfication report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94      2081
           1       0.74      0.61      0.66       385

    accuracy                           0.90      2466
   macro avg       0.83      0.78      0.80      2466
weighted avg       0.90      0.90      0.90      2466

Confusion matrix:
 [[1997   84]
 [ 152  233]]
```

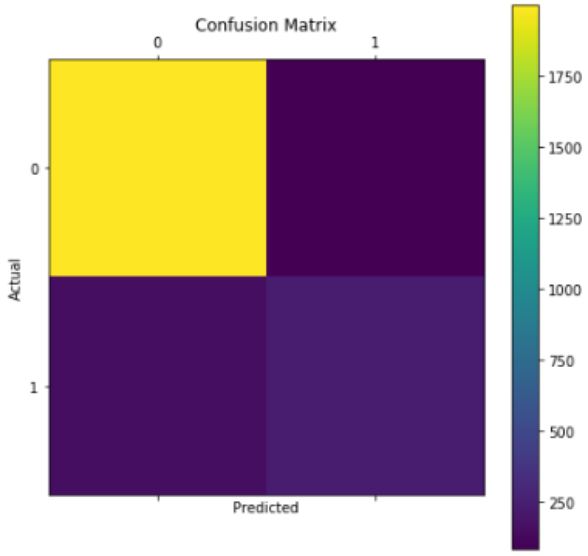


Fig-9: Decision Tree Classification Report and Confusion Matrix

3. Naïve Bayes Classifier

```
The classification report is:
              precision    recall  f1-score   support

           0       0.93      0.77      0.84      2081
           1       0.36      0.71      0.48       385

    accuracy                           0.76      2466
   macro avg       0.65      0.74      0.66      2466
weighted avg       0.84      0.76      0.78      2466

The score of model on test data is:  0.7570965125709651
```

Fig-10: Naïve Bayes Classification Report

4. Random Forest Classifier

We also test different parameters for Random Forest classifier, criterion, max_depth and min_samples_split, to find the fitted model.

```
Training Accuracy :  0.9965531224655312
Testing Accuracy :  0.897404703974047
--------------------------------------------------------------------
[0.92051906 0.90510949 0.88969992 0.87347932 0.87429035]
Overall Accuracy:  0.89  (+/- 0.04)
```

Fig-11: Accuracy of Random Forest Classifier (criterion=entropy)

```
Training Accuracy :  0.9965531224655312
Testing Accuracy :  0.9006488240064883
--------------------------------------------------------------------
[0.91727494 0.9136253  0.88888889 0.87104623 0.87388483]
Overall Accuracy:  0.89  (+/- 0.04)
```

Fig-12: Accuracy of Random Forest Classifier (criterion=gini)

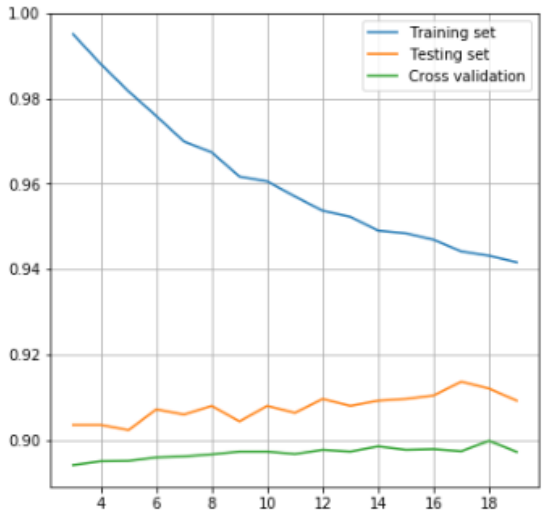


Fig-13: Accuracy of Random Forest Classifier (criterion=gini)

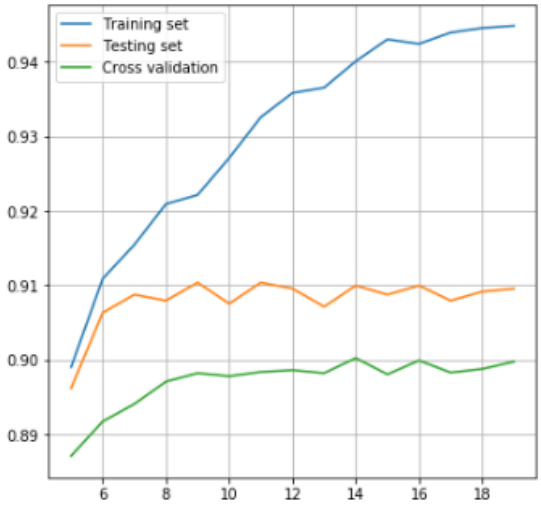


Fig-14: Accuracy of Random Forest Classifier (criterion=gini, min_samples_split=17)

Then we fit the final model by using the above parameters:

criterion = gini

min_samples_split =17
max_depth = 9

```
Training Accuracy :  0.9225466342254663
Testing Accuracy :  0.9099756690997567
              precision    recall  f1-score   support

           0       0.92      0.97      0.95      2081
           1       0.79      0.57      0.66       385

    accuracy                           0.91      2466
   macro avg       0.86      0.77      0.81      2466
weighted avg       0.90      0.91      0.90      2466
```

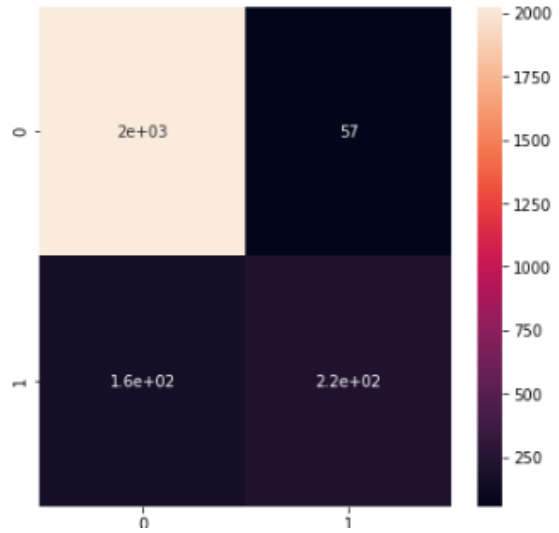


Fig-15: Random Forest Classification Report and Confusion Matrix

We apply four different classifiers, including KNN, decision tree, Naïve Bayes, and random forest, on our dataset. By comparing the accuracy score on both training dataset and testing dataset, we believe random forest classifier performs the best with a score (0.9099) on testing dataset. What's more, even for Recall, Fl-score, Random Forest Classifier performs the best. Thus, we can conclude that Random Forest Classifier is the best algorithm for this dataset.

Customer Analysis

Based on all these plots and data, we may need to recommend more related products to customers and promote our webpages through multiple ways, so that customers have more chances to see the products and increases the chance to buying these stuffs.

For our model, we analyze the online shopper intention with multiple dimensions, so based on current dataset, we are able to show how customers behave while they are visiting the webpages, maybe the reviews are not as good as they thought, or maybe they have another good deals

offered by others, etc. Thus, to increase our business sales, we can create a model to predict customers behavior during the online shopping session in the future.

In real business, returned visitors are the most likely buyers for our products, so trade discounts may be used as a competitive stratagem to secure customer loyalty. And on the other hand, some survey proves that customer experience was ranked by nearly 50 percent as the primary issue related to customer loyalty and retention. Thus, after-sales service or any other factors should be a significant factor to add more competitive for a company.

Conclusion

Form all above information, we have studied the performance of different supervised learning algorithms on the empirical data of online shoppers. The goal of the work was to identify a suitable model which can predict the purchase intention of a shopper visiting the webpages of certain online shop more accurately. For this purpose, we have run different classification algorithms on a diversified data set containing the historical data of online shoppers. From our experiments, we have found Random Forest as more suitable than others in terms of different evaluation metrics.

Though we have got a decent accuracy for this diversified dataset where data points are very sparse, the accuracy can be increased by using deep learning. Besides, more data set can also be used to observe though in that case, the accuracy will increase as we have experimented very sparse dataset. At last, we can say that our finding of using Gradient Boosting and Random Forest for predicting the purchase intention of online shoppers can make e-commerce more easy, profitable and comfortable for all the stakeholders.