**Online Shoppers Purchasing Intention**

# Abstract

In this project, I will work on the Online Shoppers Purchasing Intention Dataset which is a two-class dataset and the target class is “Revenue”. The dataset consists of 12330 data points and 18 attributes. The dataset is originally published at UCI machine learning Repository (Sakar, 2018). Out 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. The goal is to predict which user will end up buying something from the portal and generate the revenue. To analyze this, I will use Support Vector Machine(SVM) and Fisher Linear Discrimination (FLD) and Report the computational Times for both training and testing and also report the confusion matrix, and accuracy rate.

# Description Of Dataset

The dataset consists of 12,330 sessions which are consists of 10 numerical and 8 categorical attributes. The dataset consists of "Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product-Related Duration" represent the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories he values of these features is derived from the URL information of the pages visited by the user and updated in real-time when a user takes an action, e.g. moving from one page to another. The "Bounce Rate", "Exit Rate" and "Page Value" features represent the metrics measured by "Google Analytics" for each page in the e-commerce site. The value of "Bounce Rate" feature for a web page refers to 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. The value of "Exit Rate" feature for a specific web page is calculated as for all pageviews to the page, the percentage that was the last in the session. The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction. The "Special Day" feature indicates 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 the transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentina’s day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8. The dataset also includes an operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

# Data Processing:

## Data Preprocessing

After using the describe and dtype function of pandas, it is obvious that datatype of ***Month, VisitorType*** are object and where else ***Weekend and Revenue*** are bool type. As we all know SVM can not process objects and bool data types. Using sklearn data preprocessing function, ***LabelEncoder()***, Month, VisitorTypes, Weekend, Revenue data were converted successfully to integer data types. After converting the data, It is mandatory to check for any missing values and in the dataset and using **isnull()** function, I cheeked for any missing values.

## Outlier Detection

As we all know outlier decrease the model performances, So, I used the I**nterquartile Range** (**IQR**) to check for the outliers in the dataset. IQR is the difference between the third quantile and first quantile. Outliers when using IQR usually observations below (Q1 − 1.5x IQR) or *boxplot lower whisker* or above (Q3 + 1.5x IQR) or *boxplot upper whisker* (Badr, 2019). As a result of this **1310 observations** has been removed and cleaned successfully.

## Feature Selection:

Feature selection is an important part of choosing the proper feature for predicting or classifying dataset. I used ***Chi2 and SelectKBest*** to select the best 10 top features for the classifier.Chi2 use the relation between the feature to choose the best features and on the other hand, SelectKBest class just scores the features using a function (in this case f\_classif but could be others) and then "removes all but the k highest scoring features" (Sklearn, n.d.). After using both the sklearn Chi2 and SelectKBest, the top 10 features, ProductRelated\_Duration, PageValues, SpecialDay, Month TrafficType, Administrative\_Duration, Informational\_Duration, ProductRelated, Administrative, Informational, have selected and created new data frame for the training and testing purpose and split the “Revenue” as target class.

## Scaling and Choosing Testing and Training dataset:

After selecting the 10 best features, I use the sklearn model selection package split data into the train set(75%) and test set(25%) as per requirements which are approximately 8265(train) and 2755(test). Similar way applied to the target variables which is creating similar dimension for the target variable.

After splitting the dataset into the test and train dataset, I use the min-max scaler to scale all the features into the range of [0,1]. As values in the raw data have a different range and a result some of machine learning algorithm will not work properly without normalizing the data. So, by scaling all the data into a range increase the efficiency.

# Result:

Please see the Python code for both the SVM and FLD and the code written separately

## Testing and training time

|  |  |  |
| --- | --- | --- |
| Time | Support Vector Machine(Sec) | Fisher Linear Discriminant(Sec) |
| Training | 0.9250023365020752 | 0.6624727249145508 |
| Testing | 0.11800289154052734 | 0.01107335090637207 |

So, training and testing time SVM is higher than FLD

## Confusion Matrix

### Fisher Linear Discriminant

|  |  |  |  |
| --- | --- | --- | --- |
| Actual Class | | | |
| Predicted Class |  | P | N |
| P | 2210(TP) | 65 (FP) |
| N | 321 (FN) | 159 (TN) |

### Support Vector Machine

|  |  |  |  |
| --- | --- | --- | --- |
| Actual Class | | | |
| Predicted Class |  | P | N |
| P | 2209(TP) | 66 (FP) |
| N | 288 (FN) | 192 (TN) |

# Conclusion:

In conclusion, I can say that it takes a longer time for training and testing for SVM than FLD.On the other hand, the accuracy rate is 87% for SVM and 86% for FLD which is also higher for the SVM as well.

# References

Badr, W. (2019, 08 6). Retrieved from towardsdatascience: https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623

Sakar, C. P. (2018, 08 31). *UCI*. Retrieved from https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset

*Sklearn*. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.feature\_selection.SelectKBest.html#sklearn.feature\_selection.SelectKBest