CS F415 DATA MINING

PROJECT REPORT

Instructor In-Charge: Dr. Apurba Das

Project Title: **Data Mining for Product Marketing Analytics**

GitHub Repository: https://github.com/sai-ashwin-2001/cs-f415-project

Submitted By:

1. Sai Ashwin Kumar C 2019A4PS0628H
2. Jayant Lingamaneni 2019A7PS0005H
3. Hari Krishna Dhamodaran 2019A4PS1263H
4. Venkata Sai Adarsh Kone 2019A3PS0388H

1. Dataset Description:

This dataset(products.csv) was retrieved from Kaggle. The link to the dataset:
https://www.kaggle.com/datasets/podsyp/how-to-do-product-analytics
The dataset size is 1.2 GB and has 8471120 rows and 9 columns. The data is provided by an online store of sporting goods. On the main page of the store, users are shown 1 of 5 banners to stimulate the sales. Our purpose is to analyze the data and get useful results/rules from it.
The first 3 columns contain the order id, the user id and page id. The 4th column shows the product category displayed on the banner (sneakers, sports nutrition, accessories, clothes, company products). The 5th column shows whether the site is accessed via mobile or desktop. The 6th column has the timestamp of the action. 7th column shows whether the user clicked by the user and whether there was a purchase of the item. 8th column is the target column which has the value 1 if an item was purchased, else 0.

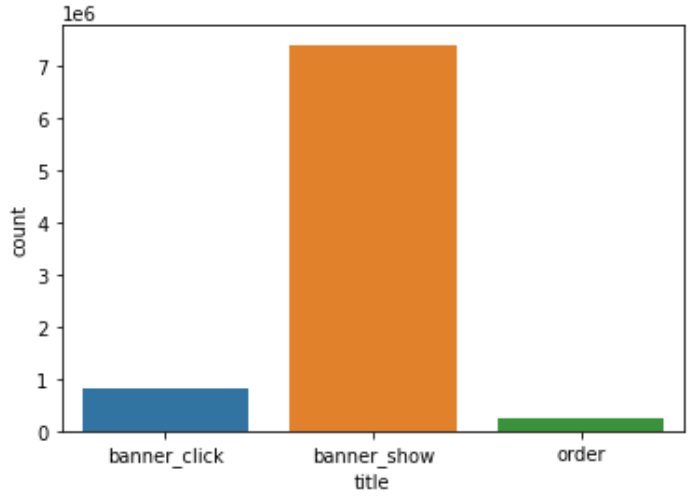
2. Problem Definition:

The dataset is from an online store of sporting goods: clothing, shoes, accessories and sports nutrition. On the main page of the store, they show users banners in order to stimulate their sales. Now one of 5 banners is randomly displayed there. Each banner advertises a specific product or the entire company. The experience with banners can vary by segment, and their effectiveness may depend on the characteristics of user behaviour. Now the job is to test the effectiveness of these banners from the available data and strategically decide when and what should the banners show.

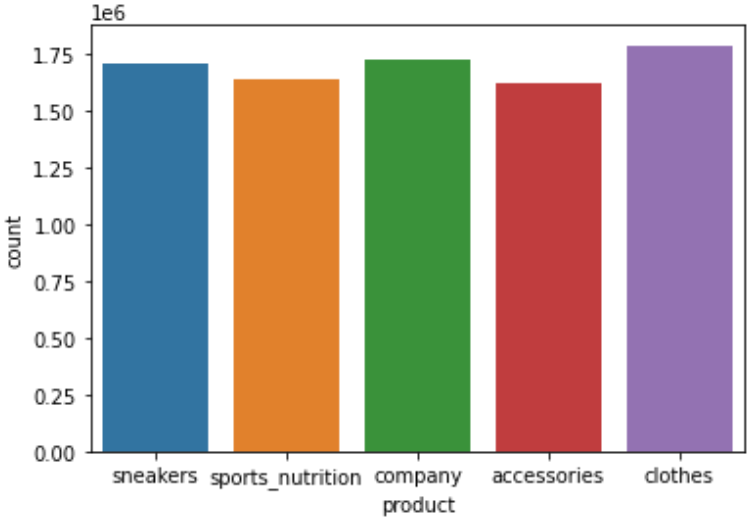
For this purpose, we have attempted classification and clustering techniques on the dataset for customer segmentation, which is the process of dividing customers into groups based on common characteristics so companies can market to each group effectively and appropriately. Then we have attempted Market Basket Analysis for analysing customer interests and buying patterns and established interesting patterns and correlations (for example, relating the time of user activity, the action of the user, product together) using associative mining methods.

3. Techniques used:

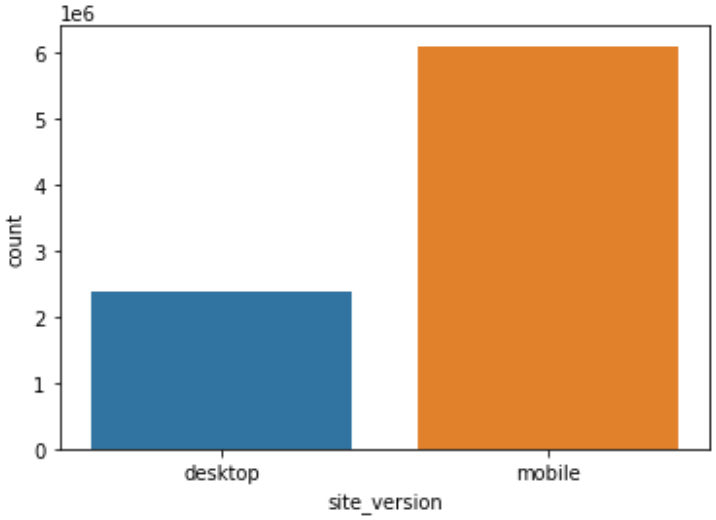
General Data Analysis and Preprocessing:



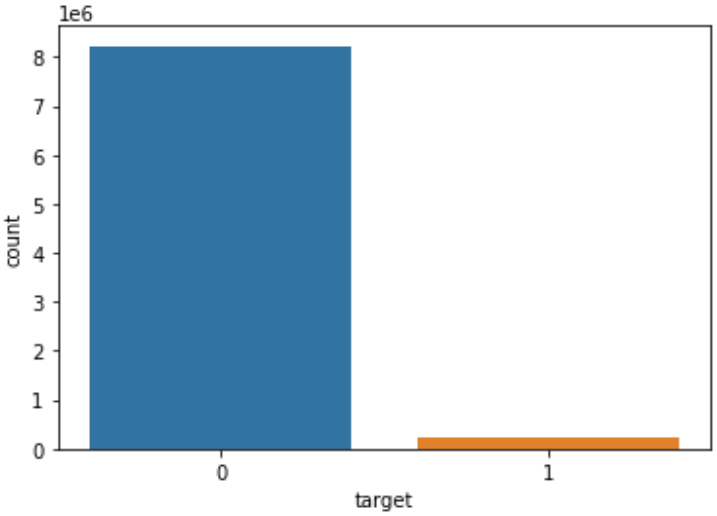
On analyzing the 7th column, we realize that for most of the datapoints the user ignores the banner when shown (banner_show) and for few datapoints the user clicks the banner and for even fewer the user orders the product.



In contrast the product column is very balanced with all products being almost equally distributed



The mobile, desktop site version split is uneven but not too heavily skewed.



As we saw in the 7th column splits, the data is heavily skewed towards no product being bought. Hence most classifiers or rules tend to just predict target as 0 to get a good accuracy. To avoid this issue, we resample the dataset such that the number of target values with 0 and 1 are equal. We end up with a dataset of 497444 rows.

In the dataset we drop the order id and page id as they don't provide any useful information regarding the task. The feature 'user_id' which is originally in a string format is converted to integers using pd.factorize which helps to get the numeric representation of an array by identifying distinct values, so that Market Basket Analysis can be performed on users' transactions. We convert the 7th column (containing banner click, banner_show,order) to a column "IsBanner_click" and convert the values into 1s and 0s. We also convert the timestamp into multiple columns corresponding to time of the day (morning, afternoon and evening and hot encode them), hour time, day of the week and the month so that the model can glean more meaningful information than it could with just a timestamp. Finally, we hot encode the 5 different product types to 0s and 1s.
The resulting dataset has 16 columns.

Following are the techniques we used for analyzing this dataset:

(i) Naïve Bayes Classifier:

We implemented the Naïve Bayes classifier using product,site_version, and time as independent attributes to predict the 'title' attribute, with a train and test split of 80 and 20. For the 'time' attribute, we processed the datetime strings and classified the times accordingly as 'Morning', 'Daylight' and 'Evening' classes. Initially we performed the naïve Bayes classifier on the unsampled dataset. It was predicting all the test values as 'banner_show' and achieving an accuracy of 87, hence we decided to sample our dataset to get equal values of both the target values. On running the classifier on the resampled data we achieved an accuracy of 66.77%.

(ii) Support Vector Machine with Principal Component analysis:

We use PCA on the resampled data to divide the X values into 2 principal components and then apply SVM with Radial Basis Function as the kernel to the principal components. We perform this just to reduce the runtime due to the high computational complexity involved. We get a low accuracy of 46% and even when attempting an SVM classification without the PCA and using lesser sample instances (again to reduce the runtime) we get an accuracy of just 50% suggesting that SVM is not a good way to classify this data

(iii) Feature Selection + Random Forest and XGB Classifiers:

We first perform 'feature selection' operation on the dataset, which is the process of reducing number of input variables to the prediction model in view of reducing computational complexity and increasing accuracy. We compute the classifier score with respect to each input variable and the features with low scores are pruned.

After feature selection is done, we apply a random forest classifier on the selected features using the GridSearchCV function of Sci-kit-learn for getting the optimal hyperparameters. We get an accuracy of 76.0% and a ROC_AUC score of 0.762

Similarly, we apply XGBoost on the selected features using GridSearchCV function of Sci-kit-learn for getting the optimal hyperparameters. We get an accuracy of 76.6% and a ROC_AUC scores of 0.767.

(iv) Clustering Analysis:

We applied K-means clustering from scratch with k value as 3 to see if the data can be clustered in a meaningful manner such that we can do appropriate customer segmentation. Since we are doing clustering on a binary data (i.e. 0s and 1s) for most of the features, taking a mean of the dataframe columnwise would represent the number of positive instances (i.e occurences of 1) of the feature with respect to each cluster. This would effectively mean the percentage of customers for whom that particular condition would hold true. From this we can infer the most popular tasks/choices of the customers in each cluster. K-means doesn't separate the clusters very well for this dataset for most of the instances, with the mean of the attributes Banner Click, products, site version being close to each other for each cluster. However, the algorithm was able to segment the customers marginally well with respect to time of action.

(v) Associative Rule Mining:

Initially we tried to do basic market basket analysis from scratch using the user ids and the product types and tried to infer association rules with the different product types. However, this resulted in very low Lift scores for all the rules, reflecting the low interestingness of the rules. We then tried the adding the is_banner_click column to the market basket analysis since the user may be interested in a product even if the user doesn't order it and just clicks on the banner. This too yielded poor Lift scores.

We then extended the associative rule mining by including all the features related to time such as time of the day (morning, evening or afternoon) and to the feature regarding the site version (mobile or desktop). Since all the attributes are categorical, we have used the dataset in the hot-encoded form. Now we randomly generate different combinations of the conditions represented by 9 columns of binary values. First we consider rules of length 2, and calculate the support, confidence and lift for all possible combinations and find the useful rules. Next we considered rules of length 3, and performed the same operation as above for different combinations. The computational complexity for the latter operation was too high that it had to run for almost an hour.

The lift scores now improved significantly and we could observe lift scores of all rules being closer to 1. We can further extract the most useful rules by applying threshold conditions on minimum support, minimum confidence and lift greater than 1 (for useful rules).

4. Source code for Implementation

The source code of implementation of the above mentioned methods can be found in the following Colab Notebook link, with each technique implementations divided into sections:

https://colab.research.google.com/drive/1SzmvBdgqKYFadF8gtuMUhFm8Y3VJdB_6?usp=sharing

5. Experimental Outcomes:

As mentioned in the above section Support Vector Machines was not suitable for this problem and the results were almost as good as random close to 0.5. Naïve Bayes classifier performed relatively better with an accuracy of 66.7% but that too is a poor result. Random Forests and XGB classifier performed better with similar accuracies slightly above 76%. Clustering analysis could segment customers only with respect to time of the day (Morning, Daylight, and Evening being preferred timings for each of the clusters) and day of the month (21st day of the month for two clusters and 7th day of the month for the other one on an average), and wasn't effective with respect to customer product buying behaviour.

Market Basket Analysis with respect to unique users and their transactions/orders wasn't that effective on this dataset where no useful rules being returned. Extended Association rule mining considering all the categorical attributes performs well having all rules whose lift values are greater than 1 thus potentially helping the store in knowing useful patterns, such as what time of the day are users more expected to buy a certain product type or which website type(mobile/desktop) may be suitable for what product type banners.

6. Our Novelty:

The previous works performed on the dataset, as obtained from Kaggle, don't contain many aspects of association analysis, rather they are more focussed on the segmentation part, which we have worked on in this project and implemented from scratch. We were also able to succesfully arrive at useful rules and correlations from the analysis.

Naive Bayes and SVM classifier has been implemented by us which was not implemented by anyone else for this dataset. Further, we have also applied ensemble/classifier combination (XG Boost and Random forest) methods that improve classification accuracy by aggregating the predictions of multiple classifiers. Though the K-means implementation was an adaptation of a previous work, we have tried to implement the algorithm from scratch without using any libraries like sklearn and were able to arrive at results comparable to the existing method.