**Abstract**

The main purpose of this project was to apply different machine learning techniques on online shoppers purchasing intention dataset and to predict whether or not the user/customer completed the purchase. Five different machine learning algorithms namely Logistic regression, Support Vector Machine , K Nearest Neighbor , Naive Bayes , Random Forest. It was observed that the highest accuracy of 89.5% was achieved using Random Forest machine learning technique.

**Introduction**

**Background**

Online shopping is a huge and growing form of purchasing and represents a huge portion of B2C (Business to Customer) revenue. 69% of Americans have shopped online at some point (1), with an average revenue of $1804 per online shopper (2). 36% of Americans shop online at least once per month! Learning how and when shoppers will research, and purchase goods online is important to businesses as they can use customer behavior insights to target advertising, marketing, and deals to potential customers to further increase their sales and revenue. The main objective of this project was to predict whether or not the user completed the purchase.

**Analysis of Dataset**

There are several steps in the analysis of any data set . Steps that were followed as a part of this project include: Data collection , Exploratory data analysis , Data preprocessing and Data Cleaning ,Splitting data features into training and testing data ,setting hyper parameters, Training and validating model and then comparing accuracy. Data was collected from the UCI repository.Dataset had 8 categorical variables and 10 numerical variables.The input and output attributes information shown in the below tables

**Table 1**

**Input Attribute Information**

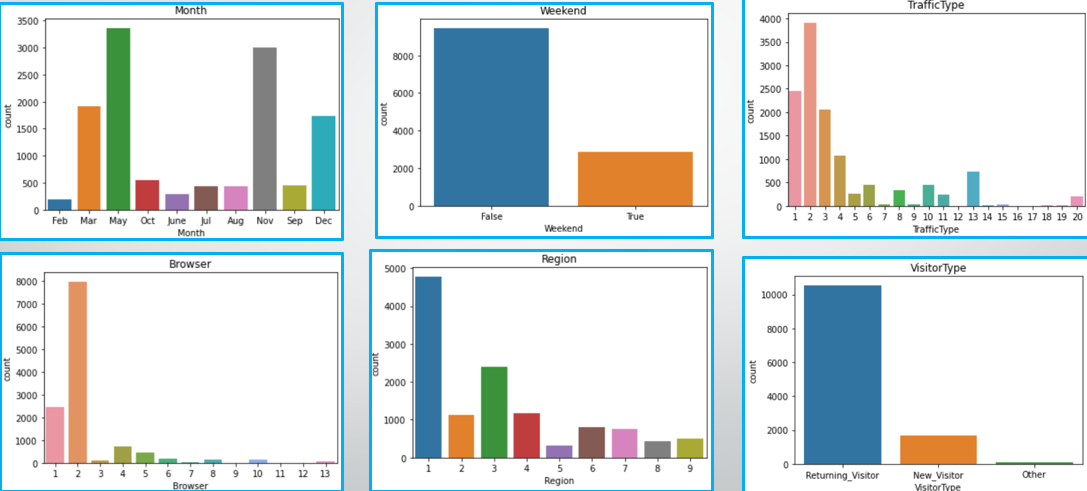
|  |  |
| --- | --- |
| Attribute | Description |
| Administrative | This is the number of pages of this type (administrative) that the user visited. |
| Administrative Duration | This is the amount of time spent in this category of pages. |
| Informational | This is the number of pages of this type (informational) that the user visited. |
| Informational Duration | This is the amount of time spent in this category of pages. |
| ProductRelated | This is the number of pages of this type (product related) that the user visited. |
| ProductRelated\_Duration | This is the amount of time spent in this category of pages. |
| Bounce Rates | The percentage of visitors who enter the website through that page and exit without triggering any additional tasks. |
| Exit Rates | The percentage of pageviews on the website that end at that specific page. |
| Page Values | The average value of the page averaged over the value of the target page and/or the completion of an eCommerce transaction. |
| SpecialDay | This value represents the closeness of the browsing date to special days or holidays (eg Mother's Day or Valentine's day) in which the transaction is more likely to be finalized. More information about how this value is calculated below. |
| Month | Contains the month the pageview occurred, in string form. |
| Operating Systems | An integer value representing the operating system that the user was on when viewing the page. |
| Browser | An integer value representing the browser that the user was using to view the page. |
| Region | An integer value representing which region the user is located in. |
| Traffic Type | An integer value representing what type of traffic the user is categorized into. |
| Visitor Type | A string representing whether a visitor is New Visitor, Returning Visitor, or Other. |
| Weekend | A boolean representing whether the session is on a weekend. |

**Table 2**

**Output Attribute Information**

|  |  |
| --- | --- |
| Attribute | Description |
| Revenue | A boolean representing whether or not the user completed the purchase. (Yes/No) |

After data collection exploratory data analysis (EDA)) was done on the data set as shown in the below figure in order to check if there any outliers in the data set

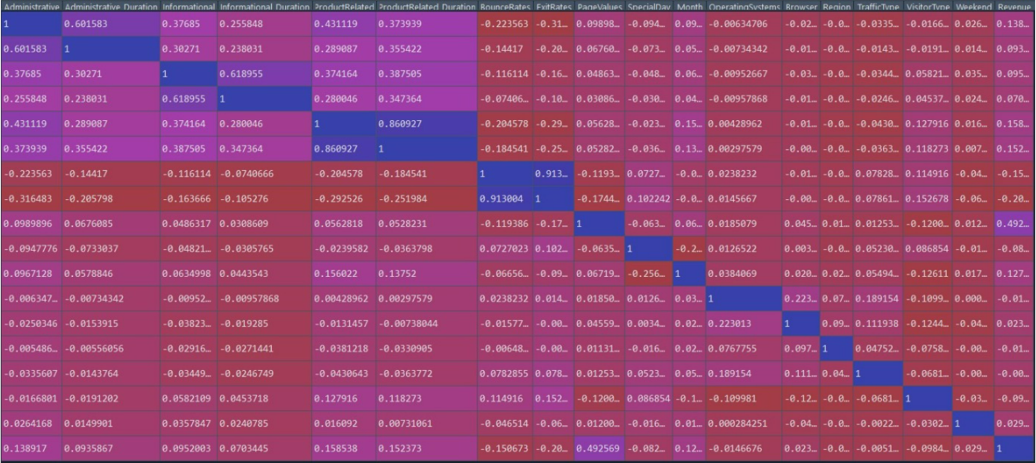


**Figure1 : Exploratory Data Analysis.**

Post EDA,Data cleaning was done in order to check if there were any missing values , duplicate values. Data set had no missing values but there were few duplicate values and those values were dropped.

Data transformation was done to Month and Visitor type attributes as these were present in string form and they were converted into integer values in order to uniformity in the data.

Then unwanted input attributes were dropped using correlation plot ( A high correlation filter - a feature selection technique). This technique was used for removing redundant features from the model and also for improving the interpretability of the model .Correlation plot shown in the below figure



**Figure 2 - Correlation Plot**

Based on the correlation plot OperatingSystems , Browser,Region,TrafficType,Weekend input attributes were removed.Then data set was split into training set and testing set ( 80% of the data is split into training data , 20% of the data is split into testing data) using hold out validation method and then K fold cross validation method is also applied on the data set for K values ranging from 2 to 32.

Post data split five different machine learning algorithms were applied to the data set followed by accuracy comparison and hyper parameter tuning.

**2. Methodology**

The project began with collection of data and the data was checked for outliers, missing values, duplicate values, data type conversions etc. The prepared data is then visualized for patterns and relationships between input features and response variables. The cleaned data is then split into a training set to fit the model and test set to check the accuracy. Five different supervised machine learning (ML) classification algorithms were fitted to the training dataset and then the accuracy of each model was analyzed in Python. The machine learning model with the least error rate is identified as the best fit model to predict the purchasing behavior of customers for an online retail store. The algorithm of each ML method is discussed in detail below:

**Logistic Regression**

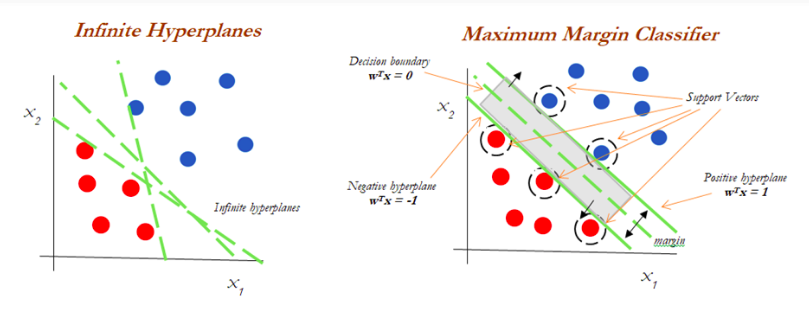
Logistic regression is a robust machine learning algorithm used for classification problems. To model a binary output variable, logistic regression employs the logistic function given below.

Logistic Function = 1/(1+)

Logistic regression is based on the concept of event "odds," which is the chance of an event happening divided by the likelihood of an event not happening. A linear relationship between dependent and independent variables is not required for logistic regression. However, independent variables must still be linearly connected to the log-odds of the outcome. Logistic Regression uses more complex cost functions.The cost function used in Logistic Regression is **Log Loss***.* Log Loss is the negative average of the log of corrected predicted probabilities for each instance. In our dataset we are predicting whether the customer has purchased or not our Response variable is categorical. So, we are using logistic regression.

**Support Vector Method**

The SVM algorithm is a supervised machine learning technique that may be used to solve both regression and classification problems. Each data observation is plotted as a point in n-dimensional space in SVM (n is the number of features you have). The hyperplane represents the decision boundaries for classifying the data points. A data point falling on either side of the hyperplane can be attributed to different classes. The hyperplane's dimension is determined by the number of input features in the dataset. The hyper-plane will be a line if we have two input characteristics. Similarly, if there are three characteristics, the plane becomes two-dimensional.The best hyperplane is the one with the greatest distance from both classes, and this is the primary goal of SVM. This is accomplished by locating many hyperplanes that best classify the labels, then selecting the one that is farthest from the data points or has the largest margin. We maximize the classifier's margin by using these support vectors. Support vectors are data points that are closer to the hyperplane and have an influence on the hyperplane's location and orientation. Its training component is simple and effective in a multi environment.



**Naive Bayes**

Naive Bayes is a classic machine learning technique which is based on the Bayes theorem on conditional probability. This classifier is mainly based on the assumption that the features are independent and are not correlated to other attributes in the data. Gaussian Naive Bayes is used for this data set. IThe algorithm works by predicting the probabilities for each class of the response variable such as the probability that a given data point belongs to a particular class. The class with the highest probability is considered as the most likely class. The Naive Bayes model is simple to construct and is especially good for huge data sets. Naive Bayes is renowned to outperform even the most advanced classification systems due to its simplicity.

**K-Nearest Neighbor**

The supervised machine learning algorithm, k-nearest neighbor, is used to address classification and regression problems. It is, however, mostly employed to solve categorization difficulties. KNN is a non-parametric, slow learning method. ‘K’ represents the number of nearest neighbors to the unknown response variable that has to be predicted. KNN works by calculating the distances between the value to be classified and all of the observations in the data and picking the K closest values, and thereby voting for the most frequent label in the case of classification. Hyperparameter tuning was done for different k values and the accuracy was estimated for each. The k value of 11 gave the highest accuracy and was chosen as the parameter value for the KNeighborsClassifier algorithm. The model is fit and the error rate/ accuracy is estimated.

**Random Forest**

Random forest algorithm is a supervised learning algorithm which can be used for both classification and regression. In the random forest algorithm, multiple decision trees are built at the same time and it uses both bagging and feature randomness to create an uncorrelated forest of decision trees. In case of classification, the output is the class selected by the majority of decision trees. For the present dataset it was observed that the prediction accuracy reached 88.6% with the default parameter setting for number of decision trees (n) to be split as 100 and the number of features considered by each tree when splitting a node (max\_features) as 1. To increase the accuracy, hyperparameter tuning was conducted to find the optimum values for n, max features and max\_depth by experimenting with different values within the range. It was observed the accuracy increased from 88.6% to 90.6% when n value is set as 100 and max features as 5. Therefore, these values are used for the final model.

|  |  |
| --- | --- |
| **Table 3**  **Hyperparameter Tuning for Random Forest** |  |
| **Hyper parameter Tuning** | **Accuracy** |
| n\_estimators= 100, max\_features = 17 | 90.1 |
| n\_estimators= 100, max\_features = 15 | 90.3 |
| n\_estimators= 100, max\_features = 10 | 90.06 |
| n\_estimators= 100, max\_features = 7 | 90.22 |
| ***n\_estimators= 100, max\_features = 5*** | ***90.63*** |
| n\_estimators= 100, max\_features = 2 | 89.13 |
| n\_estimators= 100, max\_features = 1 | 88.68 |
| n\_estimators= 10, max\_features = 7 | 89.09 |
| n\_estimators= 30, max\_features = 7 | 90.10 |
| n\_estimators= 50, max\_features = 7 | 90.26 |
| ***n\_estimators= 150, max\_features = 7*** | ***90.55*** |
| n\_estimators= 500, max\_features = 7 | 90.30 |
| n\_estimators= 1000, max\_features = 7 | 90.02 |

**Table 4**

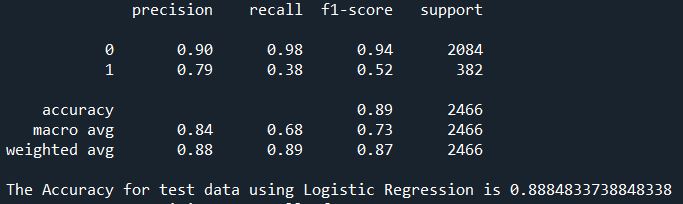
**Accuracy for different k -values in KNN**

|  |  |
| --- | --- |
| **KNN** | **Accuracy** |
| 1 | 0.843 |
| 2 | 0.8714 |
| 3 | 0.8673 |
| 4 | 0.8759 |
| 5 | 0.8742 |
| 6 | 0.8795 |
| 7 | 0.8799 |
| 8 | 0.8815 |
| 9 | 0.8815 |
| 10 | 0.8799 |
| ***11*** | ***0.8836*** |
| 12 | 0.8828 |
| 13 | 0.8824 |

## **3. Results**

**Logistic Regression**

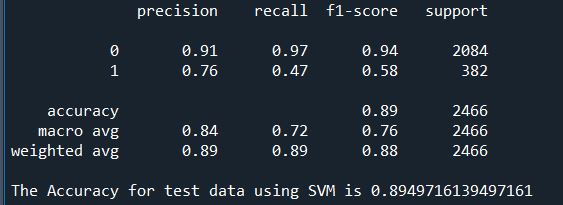
Accuracy for Logistic Regression is 88.8%



**Figure 3:Output for Logistic Regression algorithm**

**Support Vector Machine**

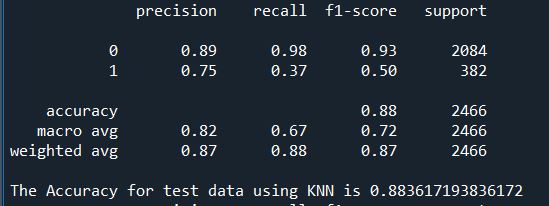
Accuracy of the SVM algorithm is 89.49%.



**Figure 4:Output for Support Vector Machine algorithm**

**K Nearest Neighbors**

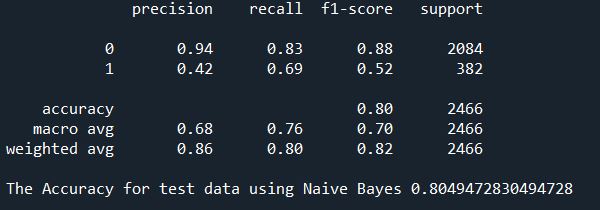
Accuracy for KNN algorithm is 88.36%



**Figure 5:Output for KNN algorithm**

**Naive Bayes**

Accuracy for Naive Bayes algorithm is 80.49% lowest among all the algorithms.

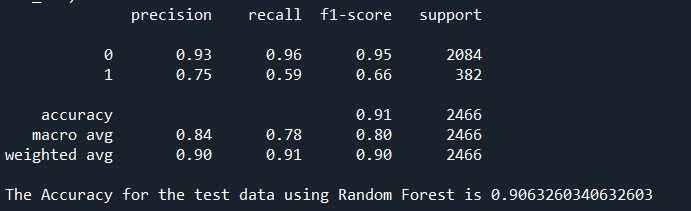


**Figure 6:Output for Naive Bayes algorithm**

**Random Forest**

Accuracy for Random forest algorithm is 89.5% highest among all the other algorithms

.



**Figure 7 :Output for Random forest algorithm**

The figures shown above depict the outputs of different machine learning algorithms used to classify revenue. Generally, accuracy score is considered to decide the performance of a machine learning algorithm and the model with highest accuracy is selected for prediction. F1 score could also be taken into consideration as it combines both precision and recall and gives us insight into how data is distributed. It is observed that the Random forest algorithm provides the best prediction accuracy for the test set among all the other algorithms.

## 

## **4. Conclusion**

The results after running multiple machine learning shows that the Random classifier has a slightly greater accuracy (90.6%) than the Support Vector Machine (89.3%) while predicting the Revenue. Random Forest showed best accuracy for 100 estimators and for 5 max features. Random forest is chosen over SVM because the response variable (Revenue) is categorical in nature.

**Table 3**

**Accuracy comparison for Different machine learning algorithms**

|  |  |
| --- | --- |
| **Analytics Model** | **Accuracy** |
| Logistic Regression | 0.888 |
| Support Vector Machine | 0.894 |
| K Nearest Neighbors | 0.883 |
| Naive Bayes Classifier | 0.804 |
| Random Forest Classifier | 0.906 |

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 