Project Documentation

**Topic**: E-commerce Shopper Purchase Prediction &Analysis

### Submitted by:

**Sankalp 102103784**

**Prabhmeet 102103785**

### Machine Learning UML 501 Submitted to:

**Dr. Ashutosh**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB**

**INDIA**

**Aug-Dec 2023**

# INDEX

1. Introduction………………………………………

2. Problem Statement………………………….........

3. Dataset…………………………………………....

4. Methodology………………………………..........

5. Result……………………………………….........

6. Conclusion……………………………………….

# Introduction

The advent of online shopping has revolutionized the way consumers interact with e-commerce platforms. Understanding user behaviour and predicting their intentions is crucial for enhancing user experience and optimizing business strategies. In the rapidly evolving landscape of online retail, marketing teams are confronted with the formidable challenge of enhancing customer acquisition, retention, and overall revenue. The key to unlocking this challenge lies in the ability to discern patterns within the vast troves of customer data generated by online shopping websites. Understanding and predicting customer behaviour can empower marketing teams to optimize promotional efforts, pricing strategies, personalization techniques, and campaign initiatives.

Marketing teams constantly strive to optimize their promotions, pricing, personalization, and campaigns to increase customer acquisition, retention, and revenue. However, identifying the most effective strategies can be challenging. Machine learning algorithms can be used to analyze past customer behavior and predict future outcomes based on various marketing strategies.

# Significance

This project seeks to address this pivotal marketing challenge through the implementation of machine learning algorithms. Specifically, our focus is on developing a predictive model capable of determining whether a customer visiting an online shopping website will culminate their visit with a purchase. By harnessing the power of historical customer interactions, this model aims to provide invaluable insights into the factors influencing purchasing decisions.

The overarching goal is to equip marketing teams with a robust tool that not only forecasts customer behavior but also serves as a guiding compass for strategic decision-making. The utilization of machine learning in this context holds the promise of revolutionizing how marketing strategies are conceived and executed, allowing for a data- driven and dynamic approach to customer engagement

# Problem Statement

Predictive Marketing Analytics for Online Retail .. . In the dynamic realm of online retail, marketing teams grapple with the intricate challenge of optimizing customer engagement and revenue generation. The vast and nuanced landscape of customer behaviour on an e-commerce platform necessitates a strategic approach to marketing initiatives. To address this challenge, we embark on a machine learning project aimed at developing a predictive model capable of forecasting whether a visitor to an online shopping website will make a purchase.

### Key Objectives:

* **Prediction of Purchase Intent:**

Develop a machine learning model that analyzes historical customer interactions on the online shopping website to predict whether a visitor will culminate their session with a purchase or not.

### Optimizing Marketing Strategies:

Provide marketing teams with actionable insights derived from the model to optimize promotional activities, pricing strategies, personalization approaches, and campaign initiatives.

### Enhancing Customer Acquisition and Retention:

Enable marketing teams to strategically target potential customers with a higher likelihood of making a purchase, thereby improving customer acquisition and retention rates.

### Data Exploration and Analysis:

Conduct exploratory data analysis to gain insights into the impact of different factors on customer behavior, such as traffic types, demographic variables, and special days.

### Addressing Imbalanced Data:

Implement techniques to handle imbalanced classes in the dataset, ensuring that the model is robust and capable of generalizing to various scenarios.

### User Interaction and Real-time Predictions:

Incorporate a user interaction component allowing for real-time predictions based on user input, providing a practical tool for marketing decision-makers.

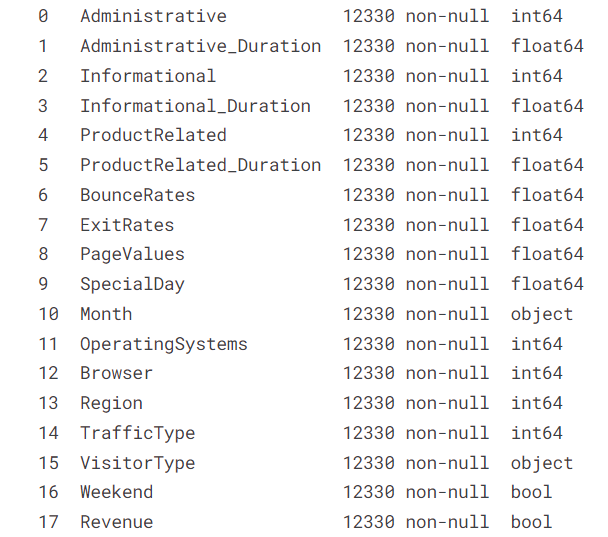
the project aims to empower marketing teams with a sophisticated tool that not only predicts customer behavior but also guides strategic decision-making for increased revenue and sustained customer engagement in the highly competitive landscape of online retail.

# Dataset

The "Online Shoppers Purchasing Intention Dataset" from UCI Machine Learning Repository is a suitable dataset for this problem statement. This dataset contains various features related to user behaviour on an online shopping website, such as the number of pages visited, the duration of the visit, and the type of traffic source. The dataset also includes a binary label indicating whether the user made a purchase or not.

This dataset is suitable for solving this problem because it provides insights into various factors that influence the purchasing decision of users on an online shopping website. By analyzing this data, machine learning models can learn to identify the most effective marketing strategies for increasing the likelihood of purchase.

* The dataset was obtained from Kaggle
* Dataset consists of 12330 samples(rows) and 18 features (columns) of which 4 are categorical and 14 are numeric
  + - Categorical : bool(2), object(2)
    - Numeric : float64(7), int64(7)



1. Administrative:Represents the number of administrative pages visited by the user.
2. Administrative\_Duration:Indicates the total time spent by the user on administrative pages.
3. Informational:Represents the number of informational pages visited by the user.
4. Informational\_Duration:Indicates the total time spent by the user on informational pages.
5. ProductRelated:Represents the number of product-related pages visited by the user.
6. ProductRelated\_Duration:Indicates the total time spent by the user on product-related pages.
7. BounceRates:The percentage of visitors who enter the site and then leave without viewing any other pages.
8. ExitRates:The percentage of pageviews on the site that end at that specific page.
9. PageValues:Represents the average value of the page(s) the user visited before completing an e-commerce transaction.
10. Special Day:Indicates the closeness of the site visit to a special day (e.g., Mother's Day, Valentine's Day).
11. Month:Represents the month of the year in which the user visited the website.
12. OperatingSystems:Represents the operating system of the user.
13. Browser:Represents the browser used by the user.
14. Region:Represents the region of the user.
15. TrafficType:Represents the type of traffic, where different numbers correspond to different sources of traffic.
16. VisitorType:Represents the type of visitor, whether New Visitor, Returning Visitor, or Other.
17. Weekend:Binary feature indicating whether the visit occurred on a weekend (1) or not (0).
18. Revenue:Binary target variable indicating whether the user made a purchase (1) or not (0).

**METHOD : CLASSIFICATION**

* In classification problems, the output is a categorical variable, and the goal is to assign each instance to one of several predefined classes or labels.
* In the case of E-commerce Shopper Purchase Prediction, the classes could be "Purchase" and "No Purchase" (1 OR 0).

# Methodology

## Data Preprocessing:

### Handling missing values and duplicates.

The code checks for missing values in the dataset using the isnull().sum() function.

If there are missing values, addressing strategies such as imputation or removal might be necessary. However, this specific code snippet does not explicitly handle missing values.

* + **Dropping irrelevant columns**. Feature Selection:

The code drops three columns ('Administrative', 'Informational', 'ProductRelated') from the dataset, likely because they were deemed unnecessary for the specific prediction task.

df = df.drop(['Administrative', 'Informational', 'ProductRelated'], axis=1)

### Encoding categorical variables.

Label encoding is applied to two categorical columns, 'Weekend' and 'Revenue,' using Scikit-Learn's LabelEncoder. This process converts categorical values into numerical representations (0 and 1).

* + - Mapping month column to num. values:

The 'Month' column, which likely contains abbreviated month names, is mapped to numerical values for better model interpretability, like 1 for January and 2 for February..

### Handling Duplicate Rows:

The code checks for and removes duplicate rows in the dataset using the duplicated() and drop\_duplicates() functions. Duplicate rows might arise from errors in data collection or recording.

### Scaling numerical features.

Min-Max Scaling for Numerical Features:

Selected numerical features are scaled using Min-Max scaling to ensure that they fall within a specific range (typically 0 to 1).

* + - The fit\_transform() method of the scaler object fits the scaler to the data and transforms the data using the scaler in one step.

In summary, the data preprocessing steps in the code encompass loading the dataset, exploring its characteristics, handling categorical variables through label encoding, mapping months to numerical values, addressing duplicate rows, and applying Min-Max scaling to selected numerical features. These steps collectively contribute to preparing the dataset for subsequent machine learning model training and analysis.

## Data Analysis :

The data analysis in the provided code involves visualizing and analyzing different aspects of the dataset to gain insights into the patterns and trends within the data. Here's a breakdown of the exploratory data analysis steps conducted:

### Traffic Type Revenue Comparison:

Grouping the data by 'TrafficType' and calculating both the average and total revenue per traffic type. This analysis aims to understand how revenue varies across different sources of traffic.

* + **Sorting and Visualizing Traffic Types by Total Revenue:** Sorting the traffic types based on total revenue in descending order and visualizing this information using a bar plot. This provides a clear representation of the impact of different traffic types on overall revenue.

### Demographic and Behavioral Analysis for Specific Traffic Types:

Filtering the dataset for two specific traffic types (Traffic Type 2 and Traffic Type 3) to conduct demographic and behavioral analyses separately.

### Demographic Analysis:

Visualizing the distribution of the 'VisitorType' for each traffic type using count plots. This provides insights into the composition of visitors based on their type.

### Special Day Analysis:

Grouping the data by 'SpecialDay' and calculating the average revenue for each special day. Sorting and visualizing the impact of special days on customer engagement using a bar plot.

### Correlation Heatmap:

Calculating the correlation matrix for all features and visualizing it using a heatmap. This provides insights into the relationships between different variables in the dataset.

### Revenue Class Distribution and Scatter Plot:

Visualizing the distribution of the 'Revenue' class using a histogram. Additionally, creating a scatter plot to explore the relationship between 'PageValues' and 'BounceRates.'

## Modeling:

### Addressing Class Imbalance using SMOTE:

The code uses the Synthetic Minority Over-sampling Technique (SMOTE) to address the class imbalance in the target variable 'Revenue.' SMOTE generates synthetic samples for the minority class (in this case, the class with less frequent occurrences) to balance the class distribution.

### Train-Test Split on the Resampled Data:

The resampled data is split into training and testing sets using the train\_test\_split function. This ensures that the class distribution is maintained in both the training and testing sets.

### LazyClassifier for Model Comparison:

The code uses the LazyClassifier from the lazypredict python library to quickly compare multiple machine learning models without specifying each one individually. It outputs a list of different classifiers along with their performance metrics

* + **Compare Models** : The code using compare\_models() from pycaret from pycaret.classification import \* library quickly compare various machine learning models.

### RandomForestClassifier:

The code uses a Random Forest classifier, a popular ensemble learning algorithm, to train a predictive model on the resampled training data. It then evaluates the model's performance on the test set using accuracy and a classification report (precision, recall, F1-score)

### Extra Trees Classifier:

Similar to the Random Forest model, an Extra Trees classifier is trained on the resampled training data. The model's performance is then evaluated on the test set, and results are displayed using accuracy and a classification report.

In summary, the modeling process involves addressing class imbalance, splitting the data, using LazyClassifier and Compare\_models for quick model comparison, and training Random Forest and Extra Trees classifiers to predict online shoppers' intentions. The evaluation metrics provide insights into the models' performance.

## Evaluation:

### Random Forest Classifier Evaluation:

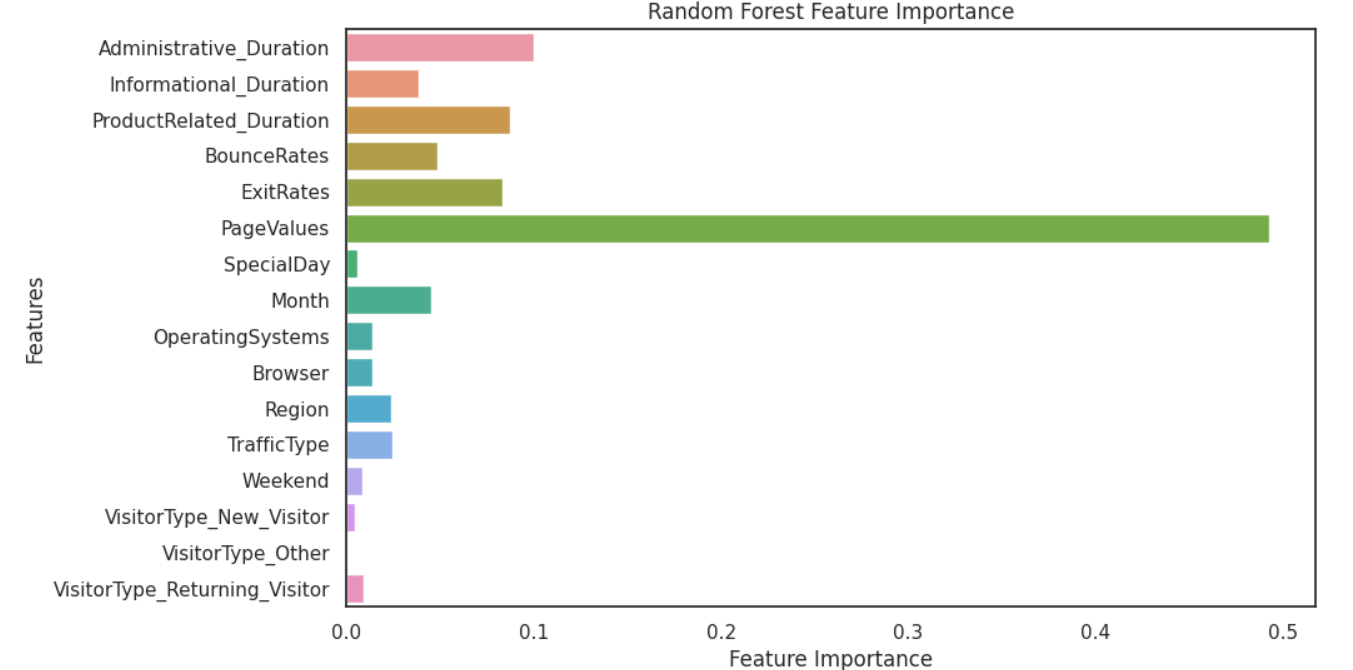
**y\_pred:** The variable holds the predicted labels for the test set using the Random Forest model.

accuracy: This metric measures the overall correctness of the model's predictions. It is the ratio of correctly predicted instances to the total instances.

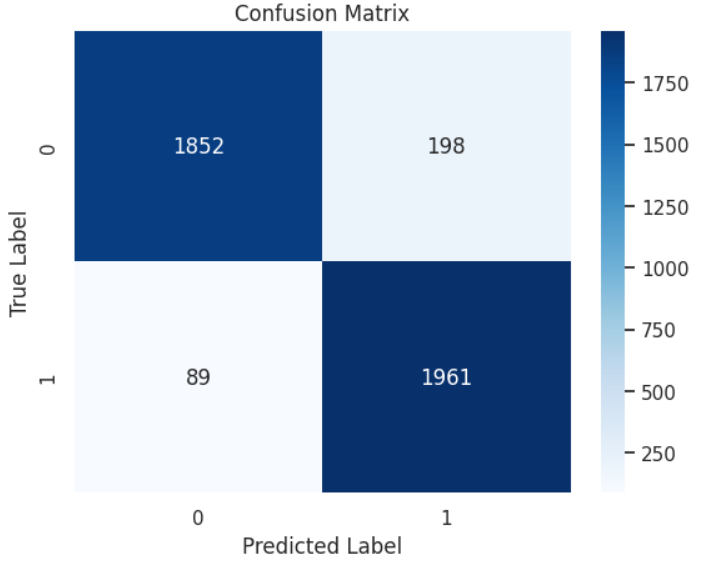
**classification\_report**: This function generates a comprehensive report containing precision, recall, F1-score, and support for each class. It provides insights into the model's performance for both positive and negative classes.

**COMPARATIVE ANALYSIS:**

1.**Feature Importance**:



**2.Confusion Matrix**:



### Extra Trees Classifier Evaluation:

**y\_pred:** The variable holds the predicted labels for the test set using the Extra Trees model.

**accuracy\_et:** This variable stores the accuracy score for the Extra Trees model.

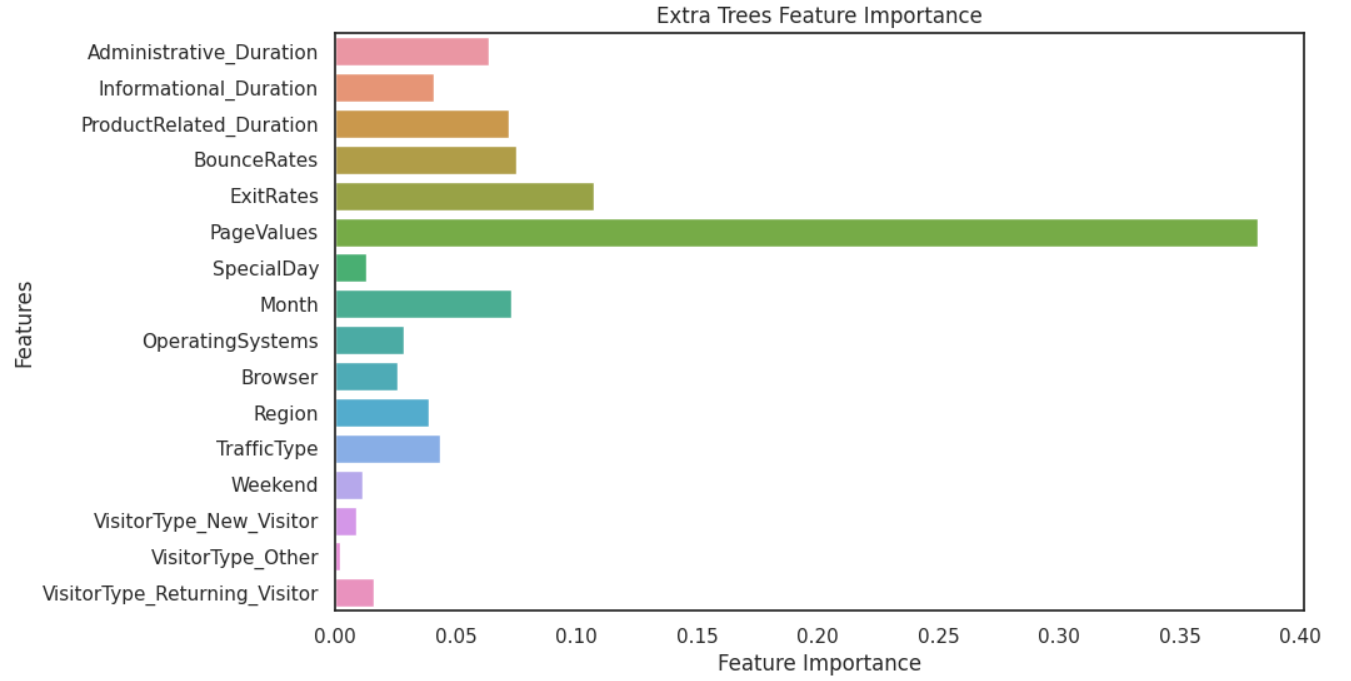
**print('Extra Trees Accuracy:', accuracy\_et):** It prints the accuracy score for the Extra Trees model.

**print(f'Accuracy: {accuracy\_et}'):** It prints the accuracy score for the Extra Trees model.

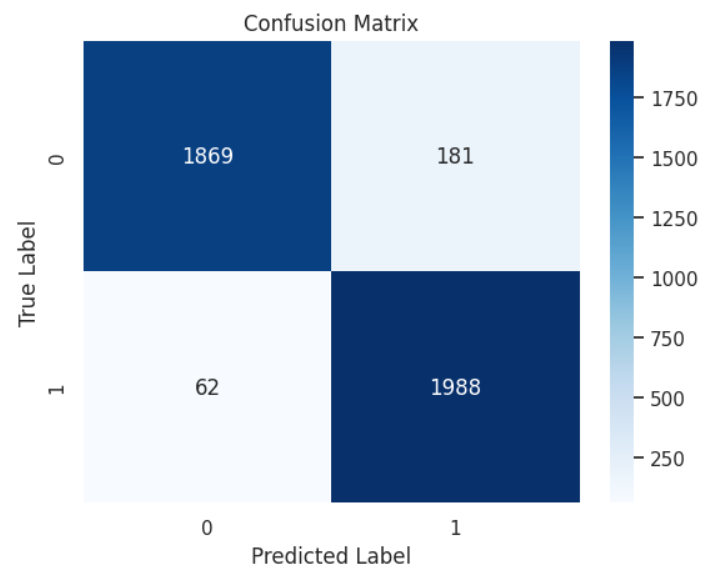
**print(classification\_report(y\_test, y\_pred)):** Similar to the Random Forest evaluation, it provides a detailed classification report including precision, recall, F1-score, and support for each class.

**COMPARATIVE ANALYSIS:**

* + **Feature Importance**



* + **Confusion Matrix**



### Confusion Matrix Visualization:

**confusion\_matrix:** This function computes the confusion matrix, which is a table showing the number of true positives, true negatives, false positives, and false negatives. The heatmap is then plotted using sns.heatmap to visualize the confusion matrix. It helps in understanding how well the model performs in terms of correct and incorrect predictions.

* + **Feature Importance Visualization**:

**feature\_importances:** feature importances are used to gain insights into which features are contributing the most to the model's performance. It's a valuable tool for feature selection and understanding the impact of different features on your machine learning model. The higher the value, the more important the feature is in making predictions.

In summary, the evaluation process involves calculating accuracy, precision, recall, F1-score, and generating a confusion matrix for both the Random Forest and Extra Trees classifiers. These metrics provide a comprehensive understanding of the models' performance on the task of predicting online shoppers' intentions.

ACCURACY:

1. EXTRA TREE CLASSIFER : 0.94
2. RANDOM TREE CLASSIFER : 0.93

# Results

The predictive models achieved approximately 94% accuracy in classifying online shoppers' intentions. The analysis revealed insights into the impact of different features on revenue, providing valuable information for targeted marketing strategies.

# Conclusion

Understanding online shopper behaviour is crucial for e-commerce platforms to optimize their marketing strategies and enhance user experience. The models developed in this project showcase the potential for accurate prediction of shopper intentions. Businesses can leverage these insights to tailor their approach and improve overall conversion rates.

INDIVIDUAL CONTRIBUTION:

DATA PRE-PROCESSING : Sankalp and Prabhmeet (80:20)

DATA ANALYSIS : Prabhmeet and Sankalp (70 : 30)

MACHINE LEARNING MODELS:

1. RANDOM TREE CLASSIFER : Prabhmeet
2. EXTRA TREE CLASSIFER : Sankalp