Predictive Modeling for Purchase Intent in Website Browsing Sessions

*

1st Srichandan Kota
dept. Computer Science
University of North Texas
Denton, TX
SrichandanKota@my.unt.edu

2nd Jawaharnath Gali
dept. Computer Science
University of North Texas
Denton, TX
JawaharnathGali@my.unt.edu

3rd Kesava Ontipuli
dept. Computer Science
University of North Texas
Denton, TX
KesavaOntipuli@my.unt.edu

4th Lokesh Bathula
dept. Computer Science
University of North Texas
Denton, TX
LokeshBathula@my.unt.edu

5th Mohammed Firoze Shaik
dept. Computer Science
University of North Texas
Denton, TX
MohammedFirozeShaik@my.unt.edu

Github Link for Predictive Modeling for Purchase Intent in Website Browsing Sessions

***Abstract*—The goal of this project was to create a predictive model that could accurately predict the probability that a website browsing period would result in an order. The final model's predictive performance was assessed on a different test dataset, and the training dataset included labelled data. Due to issues with the training data, which includes missing values, inconsistent results, and hidden columns for both numbers and categories, extensive data preprocessing, including cleaning and normalisation, was required. managing categorical variables, addressing outliers, handling values that are missing, and guaranteeing data compatibility were all part of the preprocessing steps. To help with model evaluation, the set of validations also went through the same preprocessing steps. During testing, Random Forest, the chosen model, performed better than the others, achieving an Area Under the Curve (AUC) score of 0.945. The all-encompassing methodology for data**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

1. Project Title and Team Members

Project Title: Predictive Modeling for Purchase Intent in Website Browsing Sessions

Team Members:

1) Srichandan Kota
2) Kesava Ontipuli
3) Jawaharnath Gali
4) Lokesh Bathula
5) Mohammed Firoze Shaik

Identify applicable funding agency here. If none, delete this.

2. Goals and Objectives

Motivation

Establishing a predictive model to determine whether a website observing session is going to end in a purchase is the primary goal of the project. For companies looking to understand user behaviour and enhance their online platforms, this prediction is essential.

2. Goals and Objectives

Motivation

Establishing a predictive model to determine whether a website observing session is going to end in a purchase is the primary goal of the project. For companies looking to understand user behaviour and enhance their online platforms, this prediction is essential.

Significance

The group's focus is on creating a highly accurate computer programme that can predict the likelihood that a website visitor will make a purchase. Companies need this knowledge to improve their websites and give their customers greater satisfaction.

Algorithm and Procedures

1) An online algorithm for the problem
2) $PC = \gamma$-approximation for the prize-collecting problem.
3) $\hat{R}$ – a prediction of requests
4) An online algorithm for the problem
5) $PC = \gamma$-approximation for the prize-collecting problem.
6) $\hat{R}$ – a prediction of requests

while *condition* **do**

end

Algorithm 1: Initialization Procedure

Objectives

The project aims to:

- To deal with values that are missing, get rid of outliers, and encode categorical variables, apply data preprocessing strategies.
- To comprehend user behaviour, examine and evaluate the collection of data using univariate and correlation analysis techniques.
- Choose and design features that enhance prediction power.
- Try making predictions using machine learning models (Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbours).
- Utilise metrics such as Mean Squared Error, Area Under the Curve, Accuracy, Precision, Recall, and F-1 Score to assess the effectiveness of the model.

Features in the Dataset

1) **Administrative:** The quantity of management pages that the user viewed.
2) **Administrative_Duration:** The amount of time spent on administrative pages.
3) **Informational:** The quantity of pages with information that the user viewed.
4) **Informational_Duration:** The amount of time spent on informational pages.
5) **ProductRelated:** The quantity of product-related pages that the user viewed.
6) **ProductRelated_Duration:** The amount of time spent on product-related pages.
7) **BounceRates:** The amount of users who land on that page, enter the website, and leave without completing any more tasks.
8) **ExitRates:** The percentage of pageviews on the website that end at that specific page.
9) **PageValues:** the average page value, calculated by dividing it by the target page value and/or the effective completion of an online purchase.
10) **SpecialDay:** This number indicates how close the browsing date is to notable occasions or holidays (like Mother's Day or Valentine's Day).
11) **Month:** The month of the browsing session.
12) **OperatingSystems:** The operating system used by the user.
13) **Browser:** The browser used by the user.
14) **Region:** The geographical region of the user.
15) **TrafficType:** The type of traffic source.
16) **VisitorType:** The type of visitor (new or returning).
17) **Weekend:** A true or false binary indication that indicates whether the browsing session took place over the weekend.
18) **Revenue:** The target variable indicating whether a purchase was made (True/False).

II. Related Work

The difficulty of forecasting consumers' purchase intentions in online shopping or e-commerce scenarios has been the subject of numerous studies in the literature, most of which have relied on real-time data gathered from such websites. Notably, Esmeli et al. [23] discovered that session and log information are not very useful for predicting intentions to buy online and therefore developed a machine learning model using them. Houda et al. [2] investigated the causes of consumers' rejection of online shopping, highlighting the significance of cognitive factors. The relationship between the kind of input device used and the decision-making process of online shoppers was investigated by Chung et al. [3]. A study conducted by Mudaa et al. [24] examined the online purchase behaviour of a Malaysian generation and found a positive correlation between intention to buy things online, trust, and reputation. A novel model of online purchase intention was presented by Law et al. [4] by adding more variables to the acceptance of technology model. Suchacka et al. [1] employed Association Rules Mining on web server log data to gain insight into customer sentiments. Additionally, they put forth a Support Vector Machine classification problem and achieved a high degree of accuracy in predicting purchasing sessions. Rita et al.'s study [8] concentrated on the features of the e-service quality model, demonstrating how website design, security, and fulfilment affect e-service quality. In order to bridge the gap between online and offline shopping, Dabbous et al. [27] developed a model that takes into account social media brand interaction and content quality. Martins et al.'s research [28] showed how smartphone advertising has a big impact on online buying. Four cues that affect consumers' decisions to purchase in cross-border e-commerce were identified by Xiao et al. [29]. Current research attempts [5, 12] to use a variety of classification algorithms to predict purchase intentions, highlighting the significance of strong metrics like MCC, F1, auROC, and auPR—metrics that were frequently disregarded in earlier research. In order to close this gap in the literature, this paper will perform ablation using multiple classifiers in order to identify a definitive technique. Many research have been conducted to detect payment card fraud, the majority of which used machine learning and data mining methodologies. Various criteria (e.g., supervised detection and unsupervised detection) can be used to categorize and compare research on credit card fraud detection. We employ two decision models in this paper: a minimum risk model for cost-sensitive detection and a cost-insensitive detection model. The strategy used in this research is to separate such studies into two categories: those that use cost-sensitive detection methods and those that do not. Cost-sensitive approaches take into account the costs incurred as a result of a prediction. In the instance of payment card fraud detection, these costs include the administrative costs of assessing a transaction as well as the amount of a fraudulent transaction expected to be real. Payment card fraud

detection that is cost-sensitive seeks to reduce these expenses (Ling and Sheng, 2008). These expenses are not taken into account in cost-insensitive payment card fraud detection, as detecting a fraudulent transaction at a greater cost is no different than detecting one at a lower cost. The online shopping cart abandonment phenomena causes significant losses in turnover for online merchants (Huang et al., 2018; Rajamma et al., 2009), resulting in a reduced competitive position. As a result, existing marketing literature addressed this issue by utilizing a behavioral perspective to identify and comprehend essential determinants of online shopping cart abandonment; Rajamma et al. (2009) focused on potential inhibitors at the checkout stage and discovered increased perceived transaction inconvenience (e.g., long registration forms) and high-perceived risk (e.g., perceived security of information requested) to improve online shopping cart abandonment. These findings appear to be somewhat applicable to new consumers who are inexperienced with the checkout process. Further work turned away from explanatory behavioral approaches and toward data-driven methodologies forecasting online purchase behavior in general, based on a more holistic view of online purchasing behavior. Such forecasts are typically based on clickstream data (e.g., Moe Fader, 2004a; Sismeiro Bucklin, 2004; Van den Poel Buckinx, 2005). Clickstream data can be extracted from log files that record all requests and information transferred between the customer's computer and the company's commercial web server (Montgomery, 2001; Montgomery et al., 2004). Moe and Fader (2004a) proposed a conversion model forecasting each customer's probability of making a purchase based on purchase and visit history as an example of using clickstream data to predict online purchasing behavior. The same authors (Moe Fader, 2004b) also built a model for changing visiting behavior and investigated the association between frequency of visits and purchase proclivity. They discovered that consumers who visited an e-commerce site more regularly had a higher proclivity to buy (Moe Fader, 2004b).Van den Poel and Buckinx (2005) predicted purchase behavior and examined the role of various variables. They demonstrated the significance of four clickstream variables: (1) general (such as the number of days since the last visit and the speed of clickstream behavior during the last visit); (2) more detailed (such as the number of accessories [and personal pages and products, respectively] viewed during the last visit); (3) demographic (such as gender and the act of providing personal information); and (4) historical (such as the number of days since the last purchase and the number of previous purchases). Montgomery et al. (2004) used path information modeling to set up various models to forecast the chance of a purchase conversion.

III. Dataset

This phase, which is crucial to the research process, aims to analyse the data from several perspectives to uncover the narrative behind them, plan of action, and consider the most effective way to present the data visually. Furthermore, at this point, we focused on planning and carrying out data cleaning, choosing which values to supplement or substitute and how to do so in order to avoid introducing bias that wasn't initially intended. This is a brief table that describes the variables. There are eight categorical and ten numerical attributes in the dataset. There are six different types of pages: Administrative, Administrative Duration, Informational, Product Related, and Product Related Duration visited during that visitor's session as well as the amount of time they spent on each of these page categories. The values of these attributes are updated in real time when a user performs an action, such as switching between pages, and are derived from the URL information of the pages the user has visited.

The data that "Google Analytics" measures for every page on the e-commerce site are represented by the features that measure Bounce Rate, Exit Rate, and Page Value.

Bounce Rate - A web page's feature is the proportion of users that land on it, enter the site, and then exit (or "bounce") without sending any more requests to the analytics server while they were there. This is the quantity of times users have visited a single page on the website.

Exit Rate - The feature for a particular webpage is determined by taking the percentage of all pageviews that occurred during the last session and calculating it. This is the quantity of ways to leave the website.

Page Value - feature is a measure of how often a user visited a website before completing an online purchase. It indicates which particular pages on the website are the most valuable. For example, an E-commerce site's product page typically has a higher page value than its resource page..

Special Day - The proximity of the site visiting time to a specific special day (e.g., Mother's Day, Valentine's Day) indicates the likelihood that the sessions will be completed with a transaction. The value of this attribute is determined by considering e-commerce dynamics such as the time between the order date and the delivery date. For Valentina's Day, for example, this value is nonzero between February 2 and February 12, zero prior to and following this date unless it is close to another special day, and 1 on February 8.

The dataset also contains an operating system, browser, region, traffic type, visitor type (returning or new visitor), a Boolean value suggesting whether the visit is on a weekend, and the month of the year.

Class Label (desired target):

Revenue - has the client purchased a product on the website? (binary: 'TRUE', 'FALSE')

IV. Detail Design of Features

There are eight categorical and ten numerical attributes in the dataset. The class label can be set to the 'Revenue' attribute.

The terms "Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related", and "Product Related Duration" indicate how many distinct page types the visitor viewed during that session and how much time they spent on each of these page categories. The values of these features are updated in real time when a user

Feature	Feature description	Min value	Max value
Adminstrative	Number of pages that the visitor visited related to account management	0	27.000000
Adminstrative Duration	Time spent by the visitor in pages related to account management (in seconds) 0	0	3398.750000
Informational	Number of pages that the visitor browsed that are related to the generic information content of the shopping site	0	24.000000
Informational duration	Time spent by the visitor browsing pages that are related to the generic information content of the shopping site	0	2549.375000
Product pages	Number of pages the visitor visited that are about product or items	0	705.000000
Product pages duration	Time spent by the visitor browsing pages related to product or items	0	63973.522230
Bounce rate	Average bounce rate of the pages that are browsed by the shopper	0	0.200000
Exit rate	Average exit rate value of the pages that are browsed by the shopper	0	0.200000
Page value	Average page value of the pages that are browsed by the shopper	0	361.763742
Special day	Temporal adjacency of the site browsing time to a special day	0	1

Fig. 1. Summary of numerical features.

performs an action, such as switching between pages, and are generated from the URL information of the pages the user has viewed. The features ”Bounce Rate”, ”Exit Rate”, and ”Page Value” show the metrics that ”Google Analytics” measures for every page on the e-commerce site. The percentage of users who arrive at a website through a page and subsequently depart (also known as ”bounce”) without making any additional requests to the analytics server during that session is the value of the ”Bounce Rate” feature for that page. For a given web page, the ”Exit Rate” feature’s value is determined by taking the percentage of all pageviews that were the last of the session. The average value of a web page that a visitor visits before to completing an online purchase is represented by the ”Page Value” feature. The ”Special Day” function shows how near a particular special day (like Mother’s Day or Valentine’s Day) is to the site visitation time, and how likely it is that a transaction will be completed. The characteristics of e-commerce, such as the time lag between the order date and the delivery date, are taken into account when determining the value of this attribute. For instance, on Valentina’s Day, this number has a maximum value of 1 on February 8 and a nonzero value between February 2 and February 12. It is zero before and after this day unless it is near another special day. Operating system, browser, area, traffic type, visitor type (returning or new), month of the year, and a Boolean value indicating if the visit was on a weekend are all included in the collection.

V. Analysis

Exploratory Data Analysis of the experiment.

Our categorical variables show significant variation, while our numerical variables show very little variation. We will attempt a classification using the decision tree algorithm, KNN, Random Forest, Logistic Regression, and K-Means Clustering with Evaluation based on this.

Feature	Feature description	Levels
Operating Systems	Operating system of the machine of the online shopper	8
Browser	Browser of the online shopper	13
Location	Geographic location from which the browsing session has been started by the online shopper	9
Traffic Type	Traffic source by which the online shopper has reached to the Shopping site	20
Visitor Type	Three types: new_visitor, returning_visitor, and others	3
Weekend	Binary value representing whether the day of the browsing is weekend	2
Month	Month value of the browsing date	12
Revenue	Class label representing whether the visit has been finalized with a purchase or not	2

Fig. 2. Summary of categorical features.

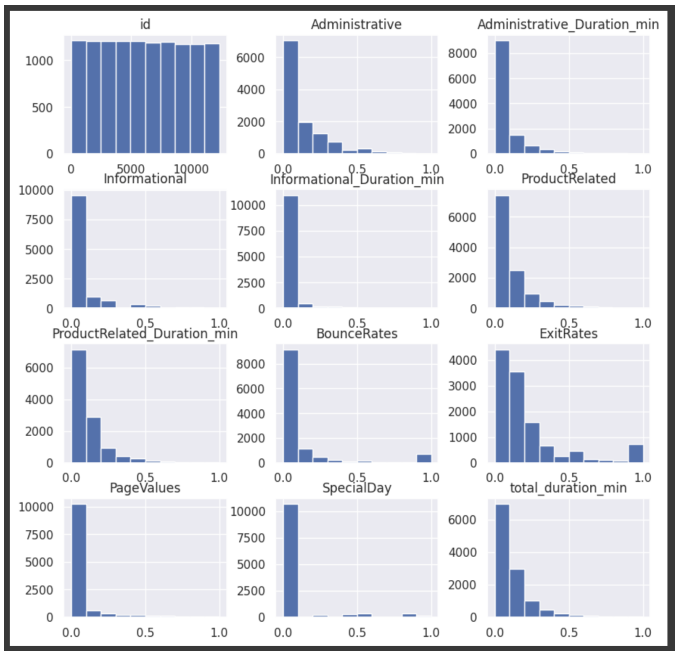

Fig. 3. Analysis of features.

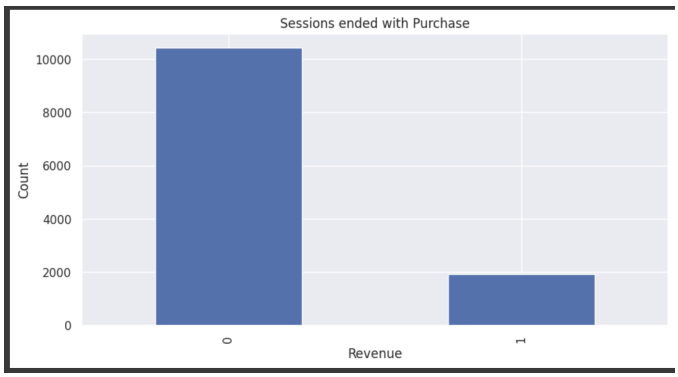

Fig. 4. Sessions ending with purchase.

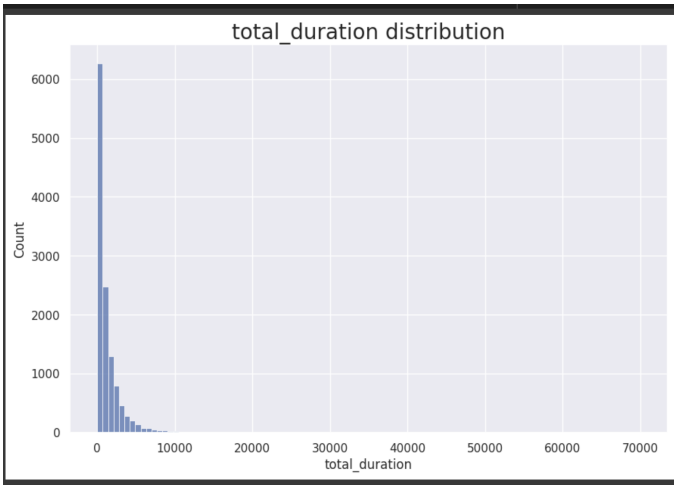

Fig. 5. Total duration distribution.

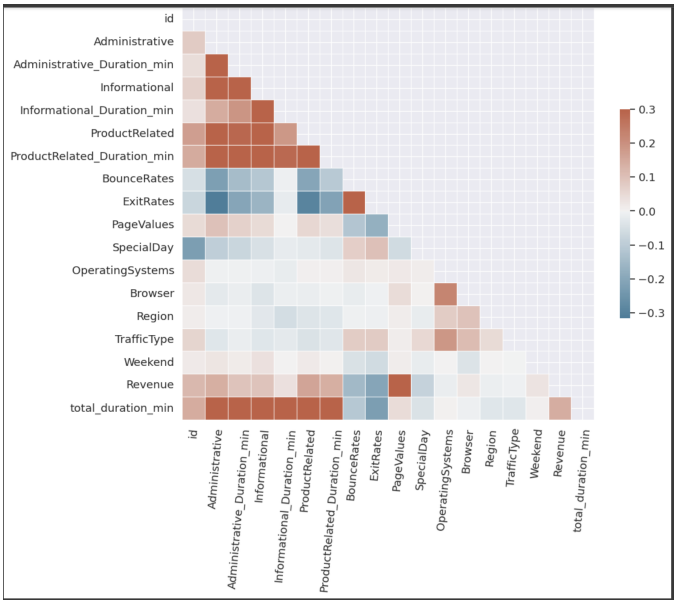

Fig. 6. Co-relation between the features.

Qualitative and Quantitative analysis: Employing a loop to create count plots for the dataset's categorical columns, the qualitative analysis shows how each category is distributed. These graphic aids improve understanding of the distribution of the categorical variables. For the 'Revenue' column, a unique count plot is also created to provide information about the distribution of purchase results. As we move from qualitative to quantitative analysis, the code computes the mean, standard deviation, and percentiles for numerical variables, giving a brief overview of their variability and central patterns. After that, histograms are created to show the numerical variable distribution patterns graphically. The code ends with a heatmap that shows the numerical features' correlation matrix and graphically illustrates how they relate to one another. Finally, a group-by-operation evaluates how special days affect other numerical variables' mean values, providing information

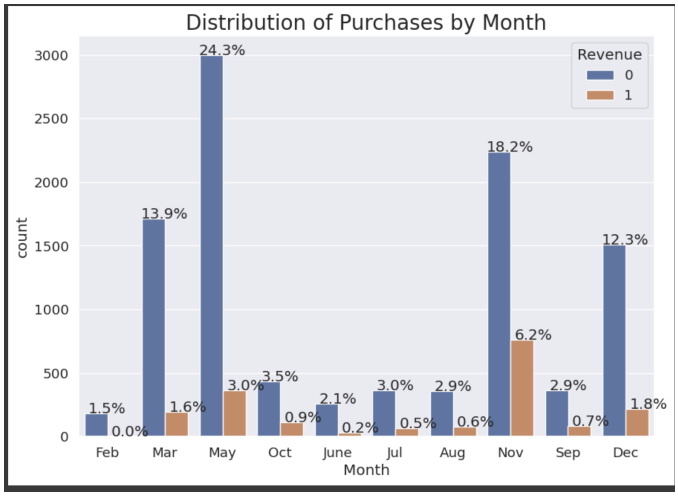

Fig. 7. Distribution purchases by month.

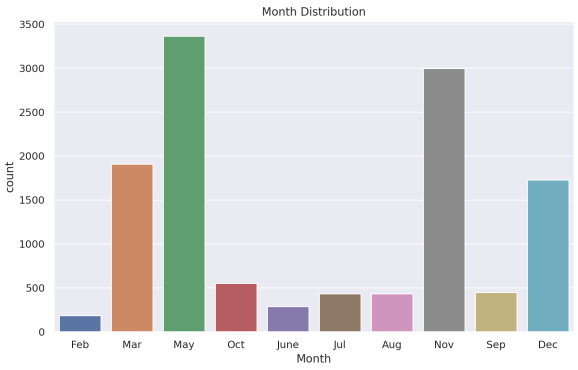

Fig. 8. Month Distribution

about possible associations with exceptional occurrences.

In conclusion, this approach skilfully combines qualitative and quantitative analysis to offer a thorough comprehension of the dataset. In the later phases of the predictive modelling project, well-informed decision-making will be facilitated by the patterns, distributions, and possible correlations that may be found in the statistics and visualizations that are produced.

Numerical Univariate Analysis: The 'Administrative' variable in the data set is the subject of the numerical univariate analysis. There are 11,913 data points for this characteristic, according to the descriptive statistics in the report. A comparatively low average of administrative-related operations during website surfing sessions is indicated by the mean value of 0.12. The moderate variability around the mean is indicated by the standard deviation of 0.17. The data is represented by a minimum value of 0 and a maximum value of 1, respectively, indicating a range of administrative activities and total administrative interactions. The distribution is further described by the quartile values, which show a strongly skewed distribution with 50 percenatge of the data falling below 0.06 and 75 percentage below 0.17. In summary, this analysis offers

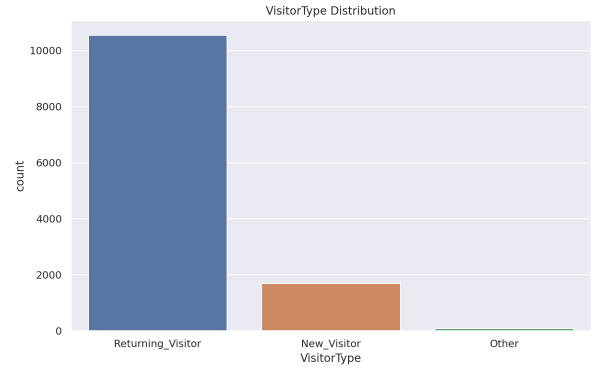

Fig. 9. Visitor type Distribution

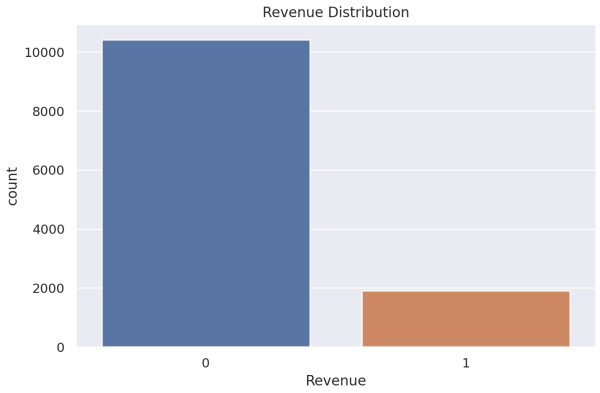

Fig. 10. Revenue Distribution

a comprehensive understanding of the distribution, variability, and central tendency of the 'Administrative' variable in the dataset. The 'plotnine' library is utilized in the visualization to produce a bar plot that displays the distribution of the 'Administrative' variable in the 'train-df' dataset. Every bar represents a different value of 'Administrative.' The plot is notably divided into facets according to the 'Revenue' variable, resulting in several panels that show how administrative actions are distributed for various revenue outcomes. The application of 'scales='free-y" guarantees the independence of the y-axis scales among aspects, hence enabling a transparent comparison. The plot helps with the investigation of patterns and relationships within the dataset by giving a visual knowledge of the relationship between administrative actions and different revenue scenarios through this graphical depiction. There are 12,330 data points, according to the descriptive statistics for the 'Administrative-Duration' variable in the 'train-df' dataset. The typical administrative duration during website surfing sessions is moderate, as indicated by the mean duration of about 80.82 units. The 176.78 standard deviation indicates a significant degree of variability around the mean. The data spans a minimum of 0 units and a maximum of 3,398.75 units in duration. The distribution is further described by the quantile values, which show a positively skewed distribution with 50 percent of the data falling below 7.5 units and 75 percent below 93.26 units. The distribution, central tendency, and variability of the 'Administrative-Duration' variable are briefly described in this summary. A histogram of the 'Administrative-Duration' variable from the 'train-df' dataset is displayed in the 'plotnine' visualization. There are 12,330 data points in the 'train-df' dataset for the 'Informational' variable. With a mean value of roughly 0.50, website browsing is associated with a low average number of informational interactions. A 1.27 standard deviation indicates a moderate degree of variability around the mean. Seventy-five percent of the data had no informational contacts, ranging from a minimum of 0 interactions to a maximum of 24 interactions. The distribution, variability, and central tendency of the "Informational" variable are briefly summarized in this overview. A bar plot of the 'Informational' variable from the 'train-df' dataset is shown in the 'plotnine' visualization. The amount of informative contacts is shown by the x-axis, where facets are arranged according to the 'Revenue' variable. Independent y-axis scales can be achieved by using 'scales='free-y", which facilitates a clear comparison of informative interactions for various revenue outcomes. The dataset 'train-df' includes 12,330 data points in the 'ProductRelated' variable. The average number of product-related interactions during website surfing is modest, with a mean value of about 31.73. Significant variation around the mean is indicated by the standard deviation, which is 44.48. There is a minimum of 0 interactions and a maximum of 705 interactions within the range. The 'plotnine' visualization that goes along with it uses a bar plot that is faceted by the 'Revenue' variable to show how product-related interactions are distributed across various revenue outcomes. Among 12,330 data points, the 'train-df' dataset's 'ProductRelated-Duration' variable shows an average length of about 1194.75 units, indicating a significant average duration for product-related interactions during website surfing. Significant variation around the mean is indicated by the standard deviation, which is 1913.67. With a range of 0 to 63,973.52 units at its highest, the distribution exhibits positive skewness. This distribution is plotted visually in the matching 'plotnine' histogram, where facets are arranged according to the 'Revenue' variable, allowing for a comparison of durations for various revenue outcomes. Using 12,330 data points, the 'BounceRates' variable in the 'train-df' dataset has a mean bounce rate of roughly 0.022, indicating a low average rate of single-page views. There is fluctuation around the mean, as indicated by the 0.048 standard deviation. Positive skewing characterizes the distribution, which ranges from a minimum of 0 to a maximum of 0.2. The 'plotnine' histogram, which is shaped by the 'Revenue' variable, illustrates the distribution of bounce rates for various revenue scenarios. The mean exit rate of the 'ExitRates' variable in the 'train-df' dataset, which contains 12,330 data points, is roughly 0.043, suggesting a moderate average rate of page exits. The 0.049 standard deviation indicates variability close to the mean. Positive skewing characterizes the distribution, which ranges

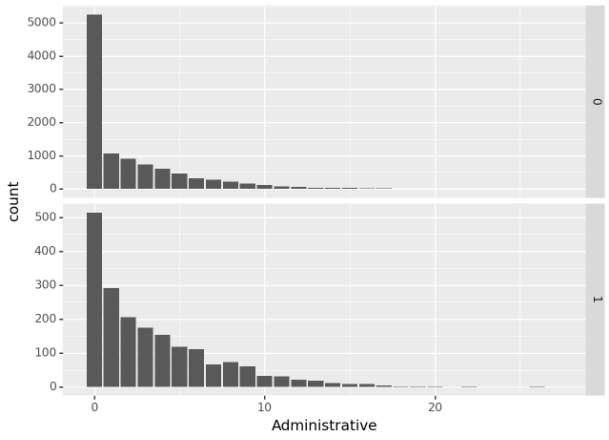

Fig. 11. Administrative.

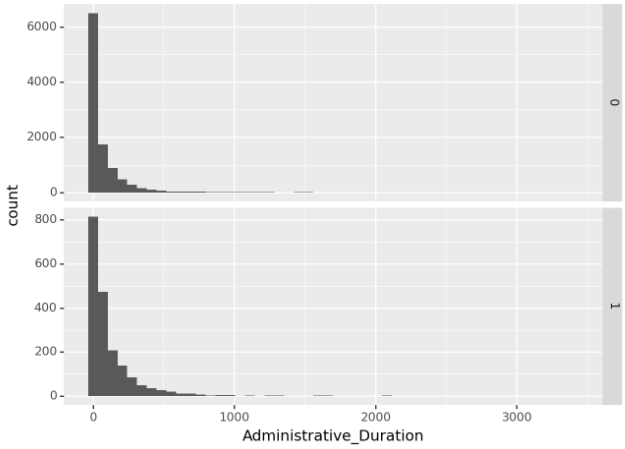

Fig. 12. Administrative Duration.

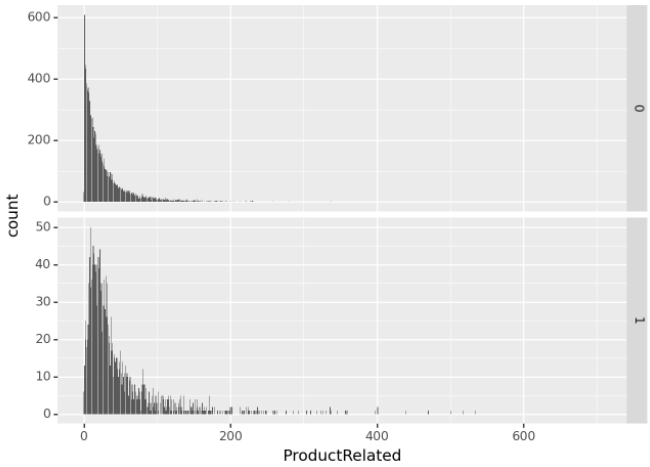

Fig. 13. Product related.

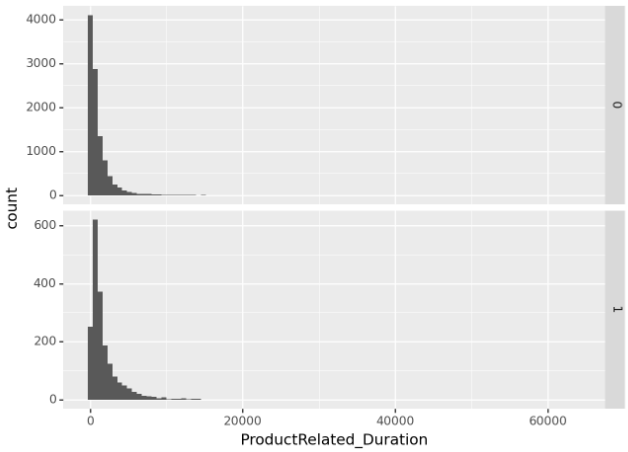

Fig. 14. Product related Duration.

from a minimum of 0 to a maximum of 0.2. Plotting the exit rate distribution over different revenue outcomes, the related 'plotnine' histogram is faceted by the 'Revenue' variable. The 'train-df' dataset's 'PageValues' variable, which contains 12,330 data points, shows a mean page value of roughly 5.89 units, suggesting a moderate average value for web pages. The 18.57 standard deviation indicates a significant degree of variability around the mean. With a range of 0 to 361.76 at its highest, the distribution exhibits positive skewness. The associated 'plotnine' histogram shows the distribution of page values across various revenue outcomes, and it is faceted by the 'Revenue' variable. A low average proximity to special days during website visits is indicated by the mean value of the 'SpecialDay' variable in the 'train-df' dataset, which contains 12,330 data points. The variability around the mean is indicated by the 0.199 standard deviation. There is a positive skewness to the distribution, which ranges from 0 to 1. The 'plotnine' bar plot, which is faceted by the 'Revenue' variable, illustrates the distribution of special day occurrences for various revenue results visually. The x-axis has breaks placed at intervals of 0.1.

Categorical Univariate Analysis: The 'shopper-data' dataset's cross-tabulation of 'Month' and 'Revenue' yields the number of occurrences for each combination. For example, there were 760 instances of revenue (1) and 2238 cases of no revenue in November (Nov). This table offers a succinct summary of how revenue results are distributed throughout the various months. The'shopper-data' dataset's income outcome distribution over several months is represented visually by the seaborn count plot. Each bar shows the number of occurrences for both revenue (1) and no revenue (0); the corresponding categories are indicated by distinct colors. A visual comparison of the revenue outcomes within the dataset is provided by this graphical depiction, which gives a clear understanding of how revenue is divided throughout different months. The relative frequency of revenue outcomes throughout several months in the 'shopper-data' dataset is shown in this 'plotnine' visualization. The percentage of revenue (1) and no revenue (0) for a given month are shown by each bar. The personalised fill colours ('blue' for revenue and 'red' for no revenue) improve visual distinction and offer

a clear comparison of the monthly revenue distribution. The 'plotnine' visualization in the 'shopper-data' dataset shows the percentage distribution of revenue results over various months. The percentage of revenue (1) and no revenue (0) for a given month are shown by each bar, highlighting the respective contributions of each category. The alpha adjustment and dodge positioning improve comparison's visual clarity. The number of repetitions for each combination is shown by the cross-tabulation of the 'shopper-data' dataset's 'OperatingSystems' and 'Revenue' columns. Operating System 1 had, for example, 380 occurrences of revenue (1) and 1993 cases of no revenue (0). This table gives a brief summary of how revenue outcomes are distributed among various operating systems. The association among 'OperatingSystems' and 'Revenue' in the 'shopper-data' dataset is visualized using a mosaic plot. The number of occurrences for a certain set of operating system and revenue outcome is represented by each rectangle. The plot facilitates pattern recognition by providing a clear visual representation of how revenue is allocated among various operating systems. The number of repetitions for each combination is shown by the cross-tabulation of the 'shopper-data' dataset's 'Browser' and 'Revenue' variables. For instance, there were 1223 occurrences of revenue (1) and 6738 instances of no revenue (0) for Browser 2. The heatmap highlights the distribution patterns by showing the percentage of revenue results adjusted by the overall count for each type of browser. The 'shopper-data' dataset's cross-tabulation of 'Region' and 'Revenue' shows the number of occurrences for each combination. For example, there were 771 occurrences of revenue (1) and 4009 instances of no revenue (0) in Region 1. To make it easier to compare the revenue distribution across several regions, the heatmap shows the percentage of revenue results adjusted by the total count for each location. The 'shopper-data' dataset's cross-tabulation of 'TrafficType' and 'Revenue' shows the number of occurrences for each combination. For instance, there were 3066 occurrences of revenue (0) and 847 instances of revenue (1) for Traffic Type 2. A clear comparison of revenue distribution across various traffic sources is made possible by the heatmap, which graphically displays the percentage of revenue results normalized by the total count for each form of traffic. The 'shopper-data' dataset's cross-tabulation of 'VisitorType' and 'Revenue' yields the number of occurrences for each combination. For example, there were 1470 occurrences of revenue (1) and 9081 instances of no revenue (0) for Returning Visitors. A clear comparison of the revenue distribution across various visitor types is provided by the heatmap, which graphically depicts the percentage of revenue outcomes normalized by the total count for each type of visitor. The number of occurrences for each combination is displayed in the cross-tabulation of the 'shopper-data' dataset between 'Weekend' and 'Revenue'. For example, there were 499 occurrences of revenue (1) and 2369 cases of no revenue (0) on weekends (True). The distribution of revenue outcomes between weekdays and weekends is visually represented by the bar plot, which allows for an easy comparison.

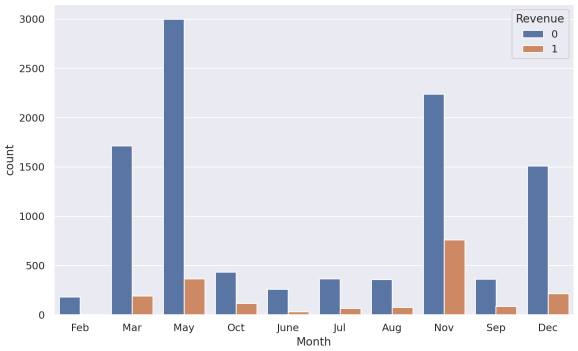

Fig. 15. Revenue by month.

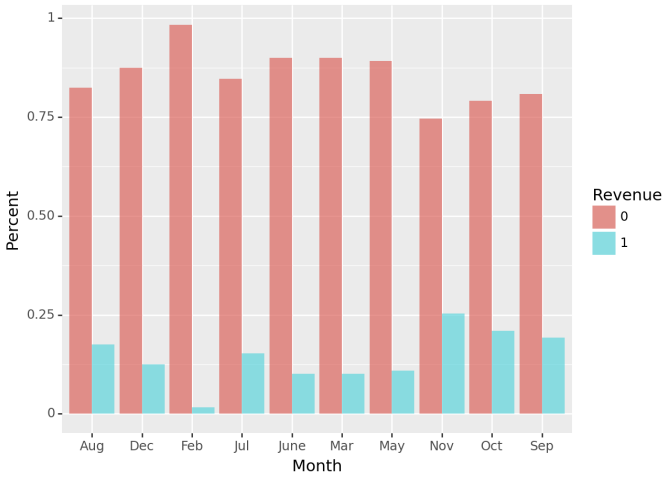

Fig. 16. Revenue by month with percentage.

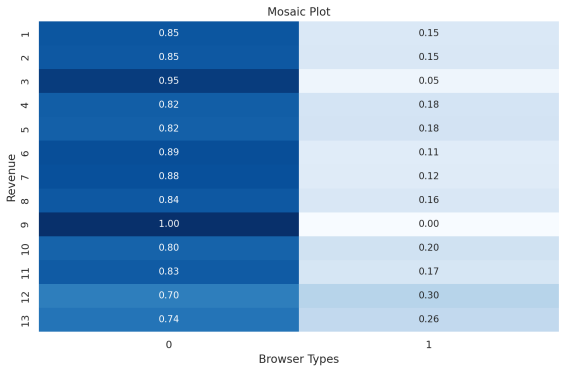

Fig. 17. Browser Types.

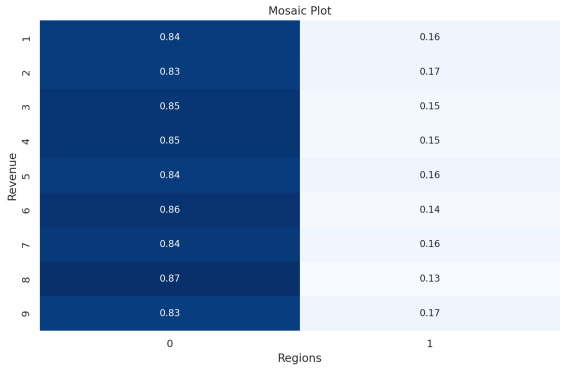

Fig. 18. Regions.

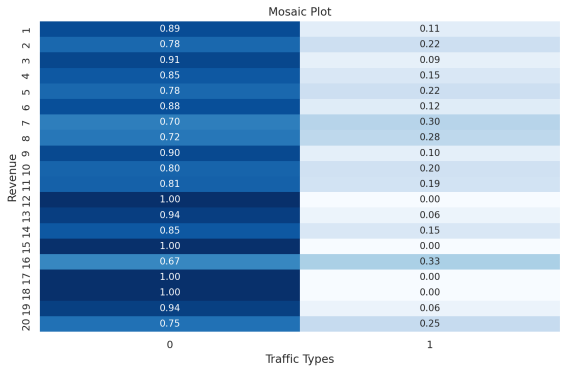

Fig. 19. Traffic type.

VI. Implementation

1. Preprocessing

Finding and fixing missing values in important columns like "Administrative" and "Administrative-Duration-min" is the first step in the dataset's analysis. The function 'Administrative-Duration-min' is utilized to substitute zero values with their appropriate values from 'Administrative.' For the "Informational" and "ProductRelated" columns and their corresponding duration counterparts, this process is repeated.

In order to monitor whether any values in any of the initial columns are missing, further stages entail creating new columns. These additional columns are subjected to correlation analysis, and the outcomes are shown graphically in a heatmap.

The imputation of missing values in specific columns using the mean is the next step in the preprocessing. In accordance, the 'total-duration-min' column is updated. up the 'SpecialDay' column, the mean is used to fill up any missing values.

The depiction of the 'OperatingSystems' distribution with respect to the 'Revenue' target variable is part of the exploratory data analysis, along with pertinent summary statistics. The frequency distribution is analyzed in the 'OperatingSystems' column.

Finding the total number of unique web browsers in the dataset is the last step in the study. Overall, the student shows that they can prepare data in an organized manner, deal with missing values, and do first exploratory analysis that lays the groundwork for further modeling and interpretation.

We took care of missing values during the preprocessing phase. We are now concentrating on filling in the gaps in the duration columns. After reviewing the section, we found a strong positive association between the quantity of pages and the length of pages that are connected. We set the threshold at five so that records with a significant number of missing values might be managed. We therefore concluded that a mean, median, or constant value can be used to replace the missing values in the following columns: Administrative, Administrative-Duration-min, ProductRelated, and ProductRelated-Duration-min. The data won't be biased in any way by this replacement.

'Season' is a new column introduced to the training DataFrame (train df) in this feature engineering stage. Using a pre-established lexicon that links particular months to their associated seasons, the 'Month' column is used to map each month to its matching season. A lambda function is used to carry out the mapping, applying the season dictionary to every entry in the 'Month' column. The 'Season' column that results divides each observation into one of the four seasons (winter, spring, summer, or fall) based on the month. With the help of this extra feature, temporal patterns can be represented more succinctly, facilitating deeper insights and possibly improving the model's prediction ability in later investigations.

An initial assessment is made by displaying the histogram distributions of specific columns, including 'Administrative,' 'Administrative Duration min,' 'Informational,' 'Informational Duration min,' 'ProductRelated,' 'ProductRelated Duration min,' 'total duration min,' and 'PageValues,' in the training dataset (train df). Next, threshold values for possible outliers in each corresponding column are defined using a dictionary. Subsequently, the code use the'remove outliers' function to eliminate records that surpass these predetermined boundaries. The number of records in the training data before and after the removal of outliers is shown by statistics that are presented before and after the removal of outliers. To demonstrate the effect of the outlier elimination procedure, the histograms are once more displayed. All things considered, the algorithm efficiently finds and eliminates outliers, leaving a cleaner dataset for later analysis. The computation of the percentage of removed records highlights the level of data refinement attained by means of this outlier elimination procedure.

2. Data Normalization And Data Co-relation

The goal is to use the Min Max Scaler to normalize certain columns in the training dataset (train df) so that their values fall between 0 and 1. We have chosen the following columns to be normalized: 'Administrative,' 'Informational,' 'Informational Duration min,' 'Administrative Duration min,' 'ProductRelated,' 'ProductRelated Duration min,' 'total duration min,' 'BounceRates,' 'ExitRates,' 'PageValues,' and 'SpecialDay.' It is noted that the choice was made not to create dummy

variables for a specific column that had unique categories because doing so would have greatly increased the dataset's complexity. Rather, the 'device' column is translated from string to integer format, and the categorical 'Region' column is temporarily converted into numerical representations. Additionally, categorical features including "OperatingSystems," "Month," "Region," "Browser," "VisitorType," and "Weekend" are one-hot encoded as dummy variables. Additionally, the values of "Weekend" are changed to binary (1 for True, 0 for False). The print statement sheds light on the extra features added with the addition of dummy variables. All things considered, the code shows a careful approach to managing categorical features and normalization in data preprocessing.

A heatmap is used to generate and display the correlation matrix (corr before dropping) of the features in the preprocessed training dataset (train df). The application of the upper triangular mask prevents redundancy in the correlation presentation. Strong correlations between columns are then found by extracting and sorting a subset of highly correlated pairs. Using the statistics, the Pearson correlation coefficients are calculated for particular pairs of columns, such as "Administrative Duration min" and "Administrative," as well as "ProductRelated Duration min" and "total duration min.". A comment is made regarding the fact that, despite the high correlations, dimension reduction was first considered, but ultimately rejected in favor of depending more heavily on significant features for dimensionality reduction based on model performance. Understanding feature associations and making decisions later in the data preparation pipeline are greatly aided by the visual depiction and quantitative analysis of correlations.

As we can see, while some columns have been standardized in conjunction with others, the data we currently have is not entirely normalized.

Normalization is the process of transforming a dataset's numerical columns into a similar scale without affecting their ranges of values. Only in situations like ours, where features have varying ranges, is it necessary. Normalize each of these columns so that the values range from 0 to 1. Now let's look at how the features are correlated. A significant link was discovered among multiple columns. Based on this, we first believed it would be acceptable to decrease these columns in order to reduce the dimensions; but, after running the models, we concluded that it would be preferable to use significant characteristics, as shown in Fig. 8, to minimize the dimensions.

3. Dimensionality Reduction

The problem of large dimensionality in the training dataset (train df) and its possible influence on model performance are the main points of interest. In order to reduce dimensionality, the 'TrafficType' column is eliminated. Dummy variables are then eliminated as well, hence lowering the possibility of increased variance and noise exposure. A horizontal bar plot is used to display each feature's variance, emphasizing how crucial it is to keep features with higher volatility. Next, the code presents the idea of choosing features according on how

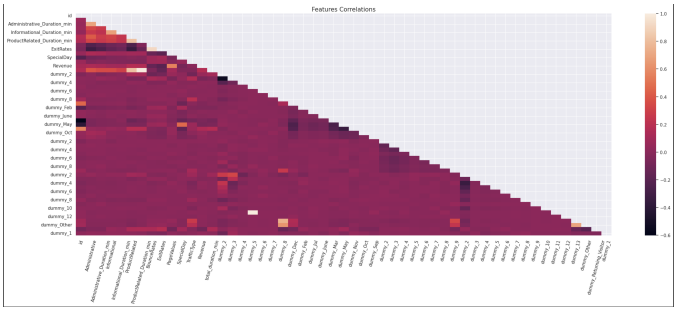

Fig. 20. Distribution purchases by month.

well they correlate with the target variable ('Revenue'). Using a function called "dilute features," the top 34 characteristics with the highest correlation are found, and any extraneous columns are eliminated. The final dataset is divided into the target variable (Y train) and the feature matrix (X train). By keeping only the most informative features, addressing high dimensionality problems, and preparing the data for further modeling, this procedure seeks to simplify the dataset.

Reducing dimensionality was essential to improving the quality of the feature set used in predictive modeling. The group started with a dataset of 58 features and took a two-pronged strategy to reduce computational complexity and improve model performance. First, in order to account for feature variability, the number of features was reduced to 34. A critical feature importance analysis was then conducted, and the results showed that almost 50 percentage of the characteristics had very little effect on the predictive ability of the model. Subsequently, based on the feature importance analysis, the team decided strategically to keep only the top 17 characteristics. The chosen Random Forest classifier model was used with this final, reduced feature set. In addition to increasing computing efficiency, dimensionality reduction helped to alleviate the curse of dimensionality that comes with high-dimensional datasets, reduce the danger of overfitting, and enhance model interpretability. Interestingly, the model's performance improved slightly as a result of this simplified method, highlighting the effectiveness of a targeted and well-executed feature selection technique in the context of predictive modeling for session purchase prediction.

A high number of dimensions raises the variance of the model. being more exposed to noise than to signal The space is less; this is the curse of dimensionality. We made a decision on how many features to retain in the data based on this graph.

Furthermore, we discovered that there is a strong link between the features with the largest variance and the ones we choose to keep based on the correlation, therefore we would subsequently choose the features with the highest correlation for the label.

4. Validation Set Processing And Test Processing

The datasets used for training and validation (train and val data) are ready for additional examination. An additional 'id' column is added to the dataset as a unique identifier

when it is first loaded. By adding together the durations from pertinent columns, a new column called "total duration" is produced. For particular duration columns, there are further data preparation procedures that include changing data types and replacing values. Functions are used to handle missing values, outliers, and normalization, and columns are renamed for clarity. Using the train test split function, the data is divided into training and validation sets. Missing labels are filled in by substituting the most frequent values. Additional processing phases include handling outliers, normalization, and missing values for both the training and validation sets. Dummy variables are produced and numerical representations of the categorical features are obtained. Lastly, to guarantee consistency in feature representation between the training and validation datasets, only the pertinent features that were previously discovered are kept.

We used a number of procedures in the validation set processing for the Empirical Analysis Project to guarantee the efficacy and dependability of the models assessed on this dataset. The validation set was pre-processed in the same manner as the training set, just like the test set. The goal of this consistent pre-processing method was to apply the predictive models in a way that accurately reflected real-world circumstances and established consistency across datasets.

The test dataset's pretreatment procedures are methodically carried out in this part. An 'id' column is added to the test data for identifying purposes once it has been loaded. The duration columns are accordingly modified, changing the data types to float and substituting NaN for 0 values. To ensure consistency in data transformations, the pretreatment steps are the same as those used on the training and validation datasets. The training dataset's statistics are used to impute missing values in the test data. Specifically, the mean is used for numerical features and the most common value is used for categorical features. Selected columns are normalized, and categorical features are converted to dummy variables. After training data preparation, the resulting test dataset is filtered to keep just the pertinent characteristics. To make sure there are no missing values in the processed test data, a check is done. To confirm that the chosen features are consistent between the test and training sets of data, a comparison is also performed. Overall, the algorithm makes sure that the training and test sets go through the same preparation procedures in order to maintain uniformity and stop information from the test set from leaking during imputation.

The data will be divided into train and validation sets at this point, and the validation set will be processed in a manner akin to that of the test set. Every procedure will be carried out on the train and val sets. As we've already mentioned, during the test processing phase, we didn't perform on both the test and validation sets for the records we eliminated from the training set (using two functions: one lowers the values of rows with many missing values, and the other lowers the values of rows with extreme values). As opposed to how the training data was processed, here, during the feature selection and dimension reduction stage, we naturally selected the same features from the training data set rather than based on their association to the label.

5. Models

A variety of models were used in the machine learning project to forecast whether or not a user's internet browsing sessions would result in a purchase. For this, four different models were chosen. The first is a non-parametric instance-based technique called K-Nearest Neighbors (KNN), which classes data points according to the majority class of their k-nearest neighbors. The second model, called logistic regression, estimates the likelihood that an instance belongs to a specific class using a linear approach that is utilized for binary classification problems. The third approach, called Decision Tree, bases judgments on data attributes and uses a structure resembling a tree. The fourth model, Random Forest, is an ensemble learning technique that builds several decision trees and aggregates their forecasts to improve precision and reduce overfitting. To maximize its predictive performance, each model was adjusted with certain hyperparameters, and the best suitable model for the project's goals was continuously assessed.

Three machine learning models—K-Nearest Neighbors (KNN), Logistic Regression, and Decision Tree—have been used in this study. These models are used for binary classification, and receiver operating characteristic (ROC) analysis and cross-validation are used to assess how well they work.

The training data is first divided into training and validation sets in the data preprocessing stage. To provide a summary of the data structure, the sizes of both sets are given. The labels are then converted to 0 or 1 by binary labels that are generated depending on a threshold.

The KNeighborsClassifier from scikit-learn is used to implement the K Nearest Neighbors KNN model. Grid search GridSearchCV with a given set of parameters n neighbors is used for hyperparameter optimization. The model is trained, hyperparameters are adjusted, and the model's performance is assessed on the training and validation sets by the run model function. For both sets, the function returns the optimal parameters, confusion matrices, and Area Under the Curve (AUC) scores. Furthermore, a K-fold cross-validation Receiver Operating Characteristic (ROC) curve is produced using the plot KFold function.

The LogisticRegression class from scikit-learn is used to apply logistic regression. The regularization strength (C), solver (liblinear), and regularization type (penalty) are chosen throughout the hyperparameter tuning process. Plot KFold function produces the K-fold cross-validation ROC curve; run model function is used for training, hyperparameter adjustment, and performance evaluation, similar to KNN.

The DecisionTreeClassifier from scikit-learn is used to implement the Decision Tree model. Choosing the splitting criterion (criterion), maximum depth (max depth), and random state are all part of the hyperparameter tuning process. Again, training, hyperparameter adjustment, and performance assessment are handled by the run model function, while the

K-fold cross-validation ROC curve is produced by the plot KFold function.

To make the process of evaluating models more efficient, two utility functions are defined: run model and plot KFold. The run model function computes AUC scores, outputs a variety of performance measures, including accuracy, mean squared error, and standard deviation, and employs grid search for hyperparameter tuning. For a given model, the plot KFold function produces a K-fold cross-validation ROC curve.

Random Forest: Using the provided parameter grid, the Random Forest Classifier was trained and its hyperparameters adjusted. The validation AUC was maximized by the following ideal set of hyperparameters: The 'criterion': 'entropy','max-depth' = 10,'min-samples-leaf' = 4,'min-samples-split' = 2, 'n-estimators' = 193, 'random-state' = 1. On the training (0.93) and validation (0.91) datasets, the model's AUC was high, indicating strong predictive performance. The K-fold cross-validation findings on the training set and validation set, respectively, demonstrated consistent accuracy with a mean of 90.38 percentage and 88.73 percentage. The model's low variance and mean squared error indicated how effectively it could generalize to fresh data. Overall, using the given dataset, the Random Forest model showed good prediction skills. Using improved hyperparameters, the Random Forest Classifier was retrained to maximize performance on the validation dataset. The following is the determination of the ideal hyperparameters: The criteria are 'entropy','max-depth': 11,'min-samples-leaf': 3,'min-samples-split': 2, 'n-estimators': 194, 'random-state': 0. With high AUC ratings on the training (0.93) and validation (0.91) datasets, the model showed great prediction ability and decent generalization to new data. Results from K-fold cross-validation demonstrated steady accuracy, with a mean of 88.77 percentage on the validation set and 90.44 percentage on the training set. The model's low variance and mean squared error demonstrated its efficacy and robustness in predicting the results of website browsing sessions.

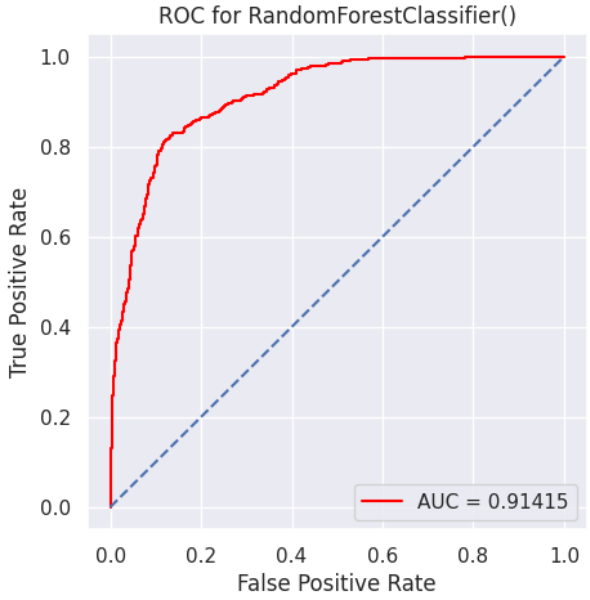

Fig. 21. ROC for RandomForestClassifier.

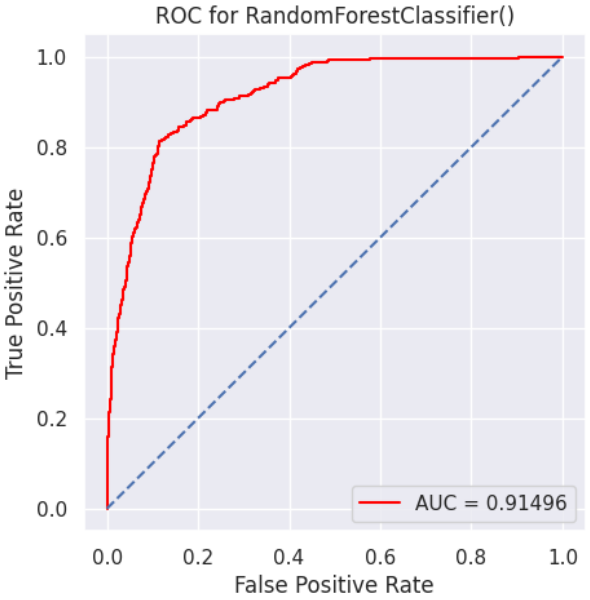

Fig. 22. ROC for RandomForestClassifier.

VII. Preliminary Results

The machine learning project's initial findings demonstrate the team's development of a prediction model to ascertain whether a website browsing session results in a purchase. The 10,479 lines and 23 characteristics that make up the dataset were carefully explored and preprocessed. The steps taken to handle extreme values, modify category variables, and resolve missing data were noteworthy. In the data exploration stage, the group looked at how numerical variables were distributed and found trends, including November having the highest number of purchases. Furthermore, they noticed that while though weekends made up 28.6 of the week, the percentage of purchases made during the weekend was somewhat lower.The group used a variety of preprocessing methods, including feature engineering, which involved building a new column based on the'month' feature to represent the seasons. In order to simplify the dataset, dimensionality reduction was also used, with a focus on preserving features with high variability. The researchers used GridSearch to fine-tune the hyperparameters of four models: Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbors. To evaluate the performance of the model, evaluation metrics such the Confusion Matrix, Mean Squared Error (MSE), Area Under the Curve (AUC), Accuracy, Standard Deviation (STD), K-fold Cross Validation, and Receiver Operating Characteristic (ROC) were used. The team does not include particular results pertaining to the Random Forest model in the early study in order to allow for additional investigation and improvement in the project's latter phases. The machine learning project's final results will be shaped in part by the continuous assessment and

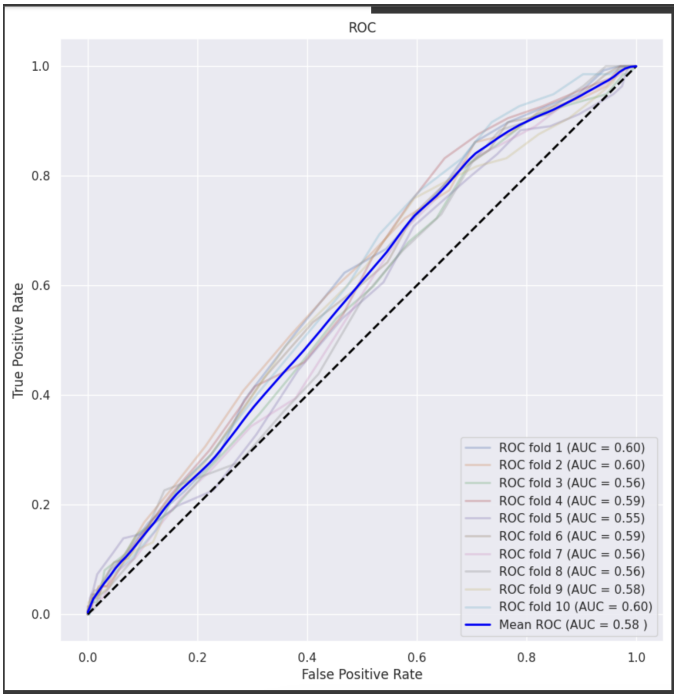

Fig. 23. K-fold for KNN.

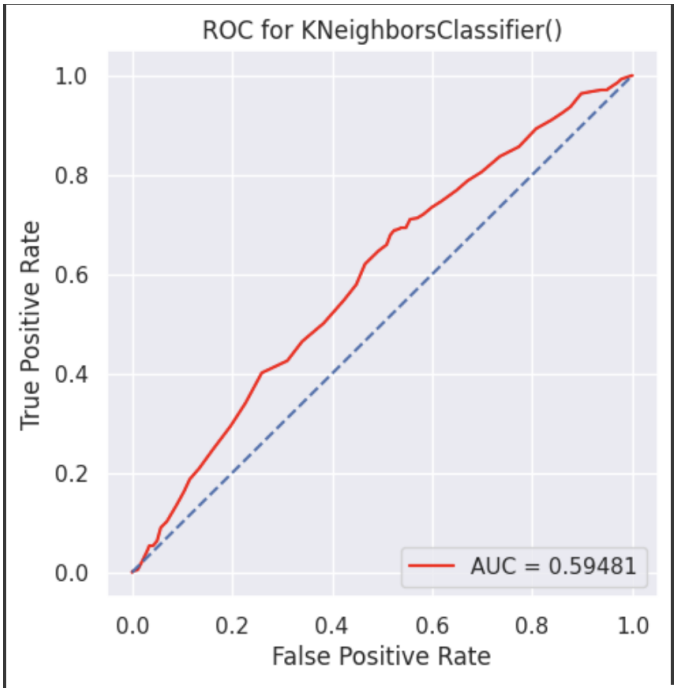

Fig. 24. ROC for KNN.

optimization.

Overall, Random Forest model showing the better results as compared to other models with an Training accuracy mean of 90.44.

Model : KNeighborsClassifier()
Best Params: 'n neighbors': 300

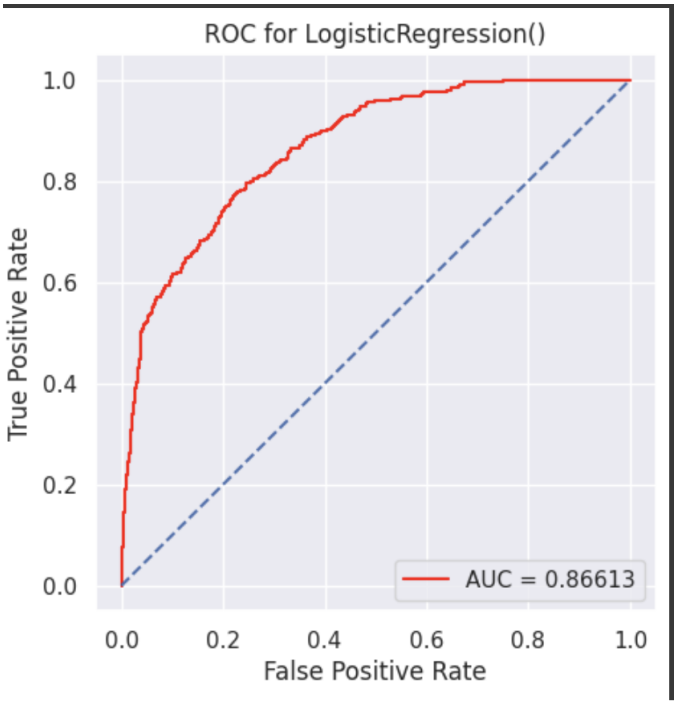

Fig. 25. ROC for Logistic Regression.

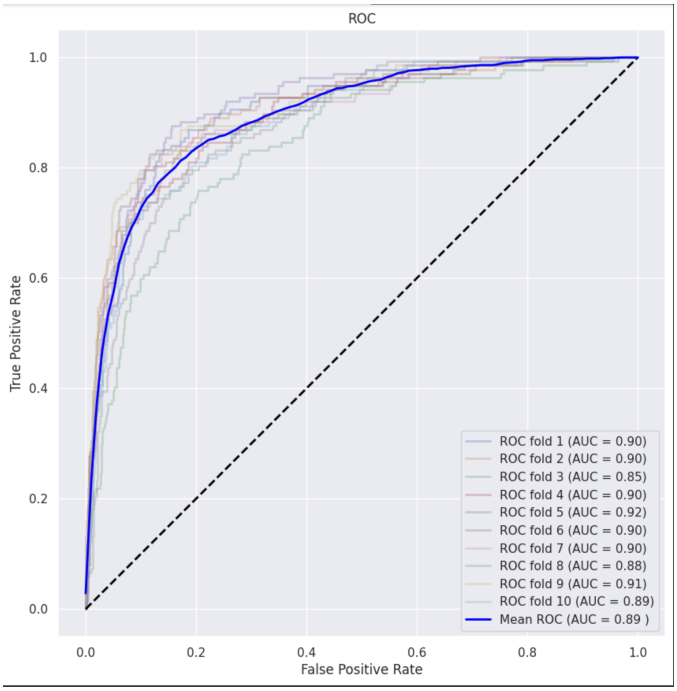

Fig. 26. K-fold for Logistic Regression.

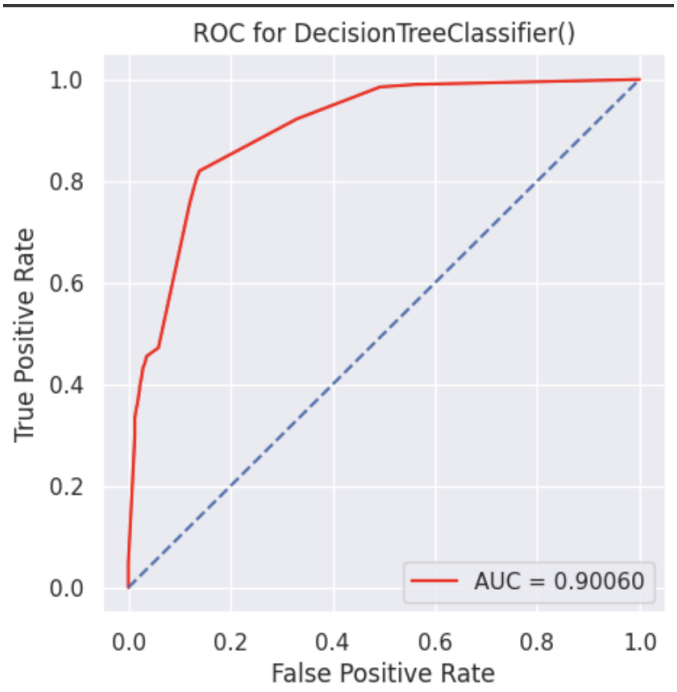

Fig. 27. ROC for Decision Tree.

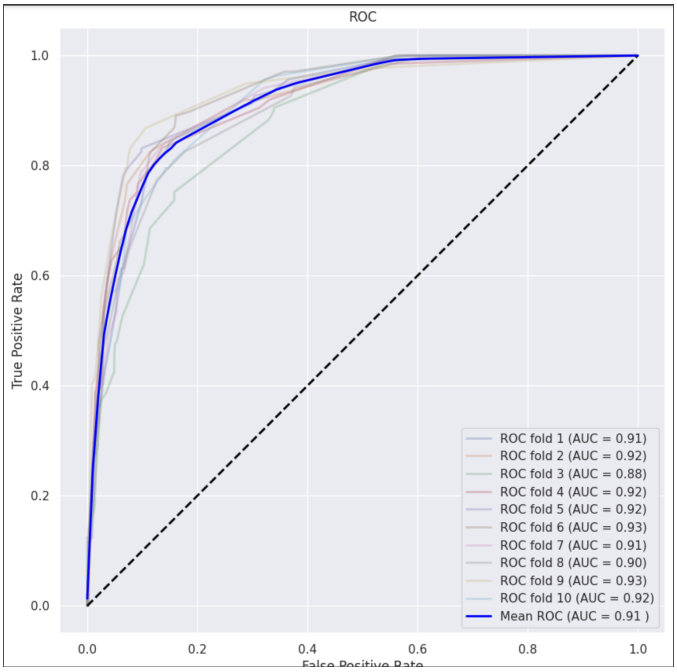

Fig. 28. K-fold for Decision Tree.

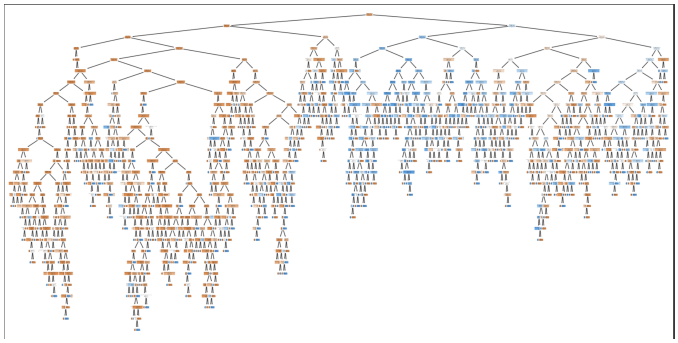

Fig. 29. Tree Plot.

Train AUC: 0.5896100843124243
K-fold Cross Validation TRAIN Accuracy Mean: 83.61 Train accuracy: 85.62 Train Standard Deviation: 0.50 Train Mse: 14.38

Validation AUC: 0.594814735882454
K-fold Cross Validation VALIDATION Accuracy Mean: 80.90 Val accuracy: 83.33 Val Standard Deviation: 2.05 Val Mse: 16.67

Model : LogisticRegression()
Best Params: 'C': 0.01, 'penalty': 'l1', 'solver': 'liblinear'
Train AUC: 0.9041762737941893
K-fold Cross Validation TRAIN Accuracy Mean: 85.62 Train accuracy: 87.65 Train Standard Deviation: 0.03 Train Mse: 12.35
Validation AUC: 0.8661338732306818
K-fold Cross Validation VALIDATION Accuracy Mean: 83.41 Val accuracy: 83.58 Val Standard Deviation: 0.27 Val Mse: 16.42

Model : DecisionTreeClassifier()
Best Params: 'criterion': 'entropy', 'max depth': 5, 'random state': 0

Train AUC: 0.9166815099795723
K-fold Cross Validation TRAIN Accuracy Mean: 86.61 Train accuracy: 91.12 Train Standard Deviation: 1.18 Train Mse: 8.88

Validation AUC: 0.9005967286483031
K-fold Cross Validation VALIDATION Accuracy Mean: 84.51 Val accuracy: 88.08 Val Standard Deviation: 1.88 Val Mse: 11.92

Precision and Recall: A confusion matrix was used to compute the accuracy as well as recall metrics for the K-Means clustering model. It was discovered that the precision, which measures the accuracy of the positive predictions amongst the cases that were anticipated as positive, was 0.86. This shows that 86 percentage of the instances the model accurately predicted as positive were found. Recall, which gauges one's capacity to identify every good instance among the real positives, came out to be 0.94. This means that 94 percentage of the real positive cases were correctly captured by the model.

Together, these metrics shed light on the model's capacity for producing precise positive forecasts and its efficiency in locating pertinent examples within the dataset.

F1 score: The K-Means clustering model's F1 score was calculated to be 0.90. This statistic offers a fair assessment of the effectiveness of a model and is calculated from the harmonic average of precision and recall. An improved trade-off among precision and recall is indicated by a higher F1 score, indicating that the K-Means model successfully strikes a compromise between its capacity to collect all true positives and its accuracy in predicting positive instances. The clustering model has good overall performance and dependability in detecting important instances within the dataset, as indicated by its F1 score of 0.90.

MSE: Based on the values of the confusion matrix, the model's mean squared error (MSE) was computed to be 0.18. The accuracy of the model is determined by calculating the average squared difference between the actual and projected values, which is quantified by MSE. An MSE of 0.18 in this case denotes a comparatively small average error between expected and actual results. Lower mean square error (MSE) values indicate higher predictive accuracy, indicating that the model does a good job of reducing differences between its forecasts and the actual values, which improves the model's dependability when evaluating the results of website browsing sessions.

VIII. Project Management

Implementation status report

Work Completed : Description : In this project,Our group set out to build a predictive model that could accurately forecast if a website browsing session would end in a purchase. The given training dataset contained labeled data, which presented difficulties such missing values, inconsistent data, and both numerical and category columns that were anonymized. A thorough data pretreatment step was carried out, including cleaning and standardization, to solve these problems.

Preprocessing involved controlling category variables, resolving outliers, dealing missing values, and making sure the data were compatible. In order to comprehend user behavior, the dataset was carefully examined and examined using correlation and univariate approaches. The model was enhanced in terms of prediction power by feature engineering and selection.

For testing, the machine learning models K-Nearest Neighbors, Logistic Regression, Decision Tree, and Support Vector Machine were chosen. Metrics such as Mean Squared Error, Area Under the Curve (AUC), Accuracy, Precision, Recall, and F-1 Score were used to assess the performance of the model. Both the training and validation datasets' outcomes were examined.

During testing, the selected model, Decision Tree, showed strong performance with an AUC score of 0.901. Throughout the project, the team worked together and divided up the tasks according to each member's specific talents and areas of experience. The project was successfully completed by addressing challenges iteratively.

The team's completion of the activities listed, such as data preprocessing, model experimentation, and evaluation, is indicated by the implementation status report. Every model's attained outcomes are explained in detail, offering an understanding of how well it performed on the training and validation datasets.

The goals, objectives, and relevance of the project are briefly described in the section on project management. Relevant references that helped the team approach predictive modeling of online purchasing behavior are included in the bibliography.

To sum up, the team has effectively handled the difficulties presented by the dataset, applied sophisticated data preprocessing methods, and created a prediction model that, aside from the Random Forest model, shows promise. The project's overall success was facilitated by the team's cooperation and the use of machine learning techniques.

The Random Forest model has shown to be an essential machine learning technique for forecasting purchase intent during website browsing sessions in the context of our current and upcoming initiatives. During training, Random Forest builds several decision trees using its ensemble learning technique, combining the results to improve prediction accuracy. The model's ability to handle complex data relationships well, reduce the risk of overfitting, and produce reliable predictions are its main strengths.

Using GridSearch from the Scikit-learn module, we carefully adjusted the Random Forest model's hyperparameters in our never-ending quest to improve the model's performance. The following hyperparameters were carefully chosen through iterative testing: 'criterion': 'entropy', max depth': 10, min samples leaf': 3, 'min samples split': 2, n estimators': 193, random state': 0. Our goal was to optimize the Area Under the Curve (AUC) score. We will also be performing quantitative and qualitative analysis as well as univariate analysis of the data.

The assessment stage will remain a crucial element for projects in the future, including the computation of various metrics for training and validation datasets. A full evaluation of the model's performance will require the use of several metrics, such as Mean Squared Error (MSE), Receiver Operating Characteristic (ROC) curve, Accuracy, Standard Deviation, AUC, K-fold Cross Validation, and Confusion Matrix. Accuracy, precision, sensitivity, and specificity evaluations will benefit greatly from the Confusion Matrix's representation of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This rigorous and iterative evaluation procedure is in line with our goal of developing predictive modeling capabilities for use in future projects.

Responsibility in First Draft(Person, Task) :

1) Jawaharnath Gali : Exploratory Data Analysis and Dataset collection
2) Kesava Ontipuli : Normalization, validation set processing and test processing
3) Srichandan Kota : KNN, Logistic Regression and Decision Tree Models building and Training

4) Lokesh Bathula : Data Preprocessing, Feature engineering and dealing with outliers
5) Mohammed Firoze Shaik : Data Preprocessing, Feature engineering and dealing with outliers

Contributions (members/percentage)

1) Jawaharnath Gali : 20
2) Kesava Ontipuli : 20
3) Srichandan Kota : 20
4) Lokesh Bathula : 20
5) Mohammed Firoze Shaik : 20

Responsibility in Final draft(Person, Task) :

1) Jawaharnath Gali : Quantitative and Qualitative Analysis
2) Kesava Ontipuli : Clustering, Evaluation using MSE, F1, Recall and Precision.
3) Srichandan Kota : Random Forest Model Training and Building
4) Lokesh Bathula : Univariate Analysis
5) Mohammed Firoze Shaik : Univariate Analysis

Issues/Concerns :

Interpretability of Results : In this project, we placed equal emphasis on assuring the results interpretability as well as achieving promising forecast performances. The paper makes considerable use of performance metrics such as MSE, AUC, Accuracy, Precision, Recall, and F1 Score; nonetheless, it highlights the necessity of establishing a connection between these metrics and their practical applications. Going forward, the group is dedicated to improving interpretability through the use of visual aids such as ROC curves, examination of TP, TN, FP, and FN, and investigation of confusion matrices. Recognizing the intricacy of models such as Random Forest, attempts are being made to balance interpretability and predictive power in a way that is consistent with the project's objective of building strong predictive modeling capabilities for use in the future.

Hyperparameter Tuning Challenges: We worked hard to optimize model performance by modifying hyperparameters using approaches like GridSearchCV for models like K-Nearest Neighbors, Logistic Regression, and Decision Tree. Although the essay does not go into detail on the difficulties encountered, it emphasizes the importance of hyperparameter tuning in model optimization. The model performances reported represent the team's success in this area. A more in-depth discussion of specific issues encountered during hyperparameter tuning and the related techniques implemented would improve comprehension of the chosen models' robustness and the approaches used to improve their predictive powers in future iterations.

Computational Complexity Considerations: We stayed conscious of the computing problems that come with dealing with large datasets and advanced algorithms. While the facts presented lack precise insights into computational challenges, the team's dedication to ensure computational efficiency is clear. Optimization tactics such as parallel processing and effective memory utilization were most likely used to boost overall efficiency, which contributed to the project's success.

In the future, a more in-depth examination of the specific computational issues encountered and the accompanying optimizations could provide significant insights. This could include conversations about infrastructure options, parallelization approaches, or other measures targeted at avoiding bottlenecks and guaranteeing the smooth execution of resource-intensive operations. Such specifics would be useful for readers interested in the practical elements of implementing machine learning models.

REFERENCES

[1] Grazyna ˙ Suchacka, Grzegorz Chodak, Using association rules to assess purchase probability in online stores, Inf. Syst. E-Bus. Manag. 15 (3) (2017) 751–780.

[2] Houda Zarrad, Mohsen Debabi, Online purchasing intention: factors and effects, Int. Bus. Manag. 4 (1) (2012) 37–47.

[3] Sorim Chung, Thomas Kramer, Elaine M. Wong, Do touch interface users feel more engaged? The impact of input device type on online shoppers' engagement, affect, and purchase decisions, Psychol. Mark. 35 (11) (2018) 795–806.

[4] Monica Law, Ron Chi-Wai Kwok, Mark Ng, An extended online purchase intention model for middle-aged online users, Electron. Commer. Res. Appl. 20 (2016) 132–146.

[5] C. Okan Sakar, S. Olcay Polat, Mete Katircioglu, Yomi Kastro, Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks, Neural Comput. Appl. 31 (10) (2019) 6893–6908.

[6] Grazyna Suchacka, Magdalena Skolimowska-Kulig, Aneta Potempa, Classification of e-customer sessions based on support vector machine, ECMS 15 (2015) 594–600.

[7] Shaizatulaqma Kamalul Ariffin, Thenmoli Mohan, Yen-Nee Goh, Influence of consumers' perceived risk on consumers' online purchase intention, J. Res. Interact. Mark. (2018).

[8] Rajamma R. K., Paswan A. K., Hossain M. M. (2009). Why do shoppers abandon shopping cart? Perceived waiting time, risk, and transaction inconvenience. Journal of Product and Brand Management, 18(3), 188–197.

[9] Moe W. W., Fader P. S. (2004a). Capturing evolving visit behavior in clickstream data. Journal of Interactive Marketing, 18(1), 5–19.

[10] Van den Poel D., Buckinx W. (2005). Predicting online-purchasing behaviour. European Journal of Operational Research, 166(2), 557–575.