CUSTOMER PURCHASE INTENT PREDICTION

USING BEHAVIOUR ANALYSIS

A PROJECT REPORT

Submitted by

NAVEEN BALAJI S (170501076)
NAVIN S (170501077)
PINAPALA ANOOP (170501086)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2021

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"CUSTOMER PURCHASE INTENT PREDICTION USING BEHAVIOUR ANALYSIS"** is the bonafide work of **"NAVEEN BALAJI S (170501076), NAVIN S (170501077) and PINAPALA ANOOP (170501086)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.R.ANITHA	**Mr.K.SRINIVASAN**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
COMPUTER SCIENCE & ENGG	**COMPUTER SCIENCE & ENGG**

Submitted for the project viva-voce examination held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The inception of ML/AI has brought many ground-breaking technological advancements in almost every field. Most of the consumer-facing companies are realising the potential of customer-level data and are finding means to utilise them its fullest potential. These data can be very crucial for companies to improve customer experience, employ sales and marketing strategies, come up with new products, recommendations etc. All these strategies hold the key to a booming company and help them attain sustainability in the volatile market. By utilizing clickstream and supplementary customer data, models for predicting customer behavior can be built. This study analyzes machine learning models to predict a purchase. These customer patterns can be used by ML/AI to provide a company with predictions on whether a certain customer will be interested in buying a specific product. In this project, we study advanced analytics tools and implement ML algorithms that predict purchase intention using customer behavioral analysis. We establish a data driven framework for predicting whether a customer is going to make a specific purchase in the near future or not. Next, to comparing models this study further gives insight into the performance differences of the models on sequential clickstream and the static customer data, by conducting a descriptive data analysis and separately training the models on the dataset.

ACKNOWLEDGEMENT

We thank our Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout our study in this college.

We express our sincere thanks to **Dr. R. Anitha,** Head of the Department, Computer Science and Engineering for her encouragement accorded to carry this project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our internal guide, **Mr. K. Srinivasan**, **Assistant Professor**, for his valuable guidance and suggestions throughout this project.

We are also thankful to our project coordinators **Dr. R.Jayabhaduri, Associate Professor, Dr. N.M.Balamurugan, Associate Professor and Ms.V.Rajalakshmi, Assistant Professor** for their continual support and assistance.

We thank our family and friends for their support and encouragement throughout the course of our graduate studies.

NAVEEN BALAJI S
NAVIN S
PINAPALA ANOOP

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iii**
	LIST OF FIGURES	**vii**
	LIST OF ABBREVIATION	**viii**
1	**INTRODUCTION**	**1**
	1.1 OVERVIEW	1
	1.2 MOTIVATION	2
	1.3 CUSTOMER BEHAVIOUR ANALYSIS	4
2	**LITERATURE REVIEW**	**6**
3	**PROPOSED WORK**	**8**
	3.1 DATASET DESCRIPTION	8
	3.2 PROPOSED ARCHITECTURE	10
	3.3 EXPLORATORY DATA ANALYSIS	11
	3.4 DATA PRE-PROCESSING	13
	3.4.1 Data Transformation	14
	3.4.2 Feature Selection	16
	3.4.3 Data Scaling	17
	3.5 MODEL SELECTION AND BUILDING	18
	3.6 MODEL EVALUATION AND METRICS	19
	3.6.1 Test Trail Split	19
	3.6.2 Metrics	20

4	**SYSTEM REQUIREMENTS**	**23**
	4.1 HARDWARE REQUIREMENTS	23
	4.2 SOFTWARE REQUIREMENTS	23
	4.3 DESCRIPTION OF TOOLS	23
5	**IMPLEMENTATION MODULES**	**28**
	5.1 EXPLORATORY DATA ANALYSIS	28
	5.1.1 Correlation Analysis	30
	5.1.2 Web Page Analysis	32
	5.1.3 Page Metric Analysis	33
	5.1.4 Visitor Analysis	34
	5.1.5 Visitor Date Analysis	35
	5.2 K-NEAREST NEIGHBOUR	36
	5.3 SUPPORT VECTOR MACHINE	38
	5.4 LOGISTIC REGRESSION	40
	5.5 RANDOM FOREST	42
	5.6 GRADIENT BOOSTING	44
6	**RESULTS AND DISCUSSIONS**	**46**
	6.1 VISUALISATION OF DATA	46
	6.2 RESULT ANALYSIS	48
7	**CONCLUSION AND FUTURE WORK**	**49**
	REFERENCES	**50**

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	Numerical Features	9
3.2	Categorical Features	9
3.3	Architecture	10
5.1	Correlational Analysis	21
5.2	Web Page Analysis	32
5.3	Page Metric Analysis	33
5.4	Visitor Analysis	34
5.5	Visitor Date Analysis	36
6.1	Feature Relation Analysis	47
6.2	Comparison Of Models After Tuning	48

LIST OF ABBREVIATIONS

ML	Machine Learning
AI	Artificial Intelligence
UCI	University Of California Irvine
EDA	Exploration Data Analysis
RF	Random Forest
LR	Logistic Regression
SVM	Support Vector Machines
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
PCA	Principal Component Analysis
KNN	K - Nearest Neighbour

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Predicting customer behavior in the context of e-commerce is becoming more important nowadays. Most of the consumer facing companies are realizing the potential of customer level data and are finding means to exploit it to its fullest potential. Starting from providing a customized home page with recommendations to providing valuable suggestions based on their browsing history, there is a plethora of information which can be derived from such customer level datasets. These datasets can be very crucial for companies to improve customer experience, employ sales and marketing strategies, come up with new products etc.

All these strategies hold the key to a booming company and help them attain sustainability in the volatile market. It increases customer satisfaction and sales, by facilitating an increase of customer experience through personalization, recommendations and special offers. Due to today's transition from visiting physical stores to online shopping, predicting customer behavior in the context of e-commerce is gaining importance. It can increase customer satisfaction and sales, resulting in higher conversion rates and a competitive advantage, by facilitating a more personalized shopping process.

By utilizing clickstream and supplementary customer data, models for predicting customer behavior can be built. This study analyzes machine learning models to predict a purchase, which is a relevant use case as applied by a large German clothing retailer. Next, to comparing models this study further gives

insight into the performance differences of the models on sequential clickstream and the static customer data, by conducting a descriptive data analysis and separately training the models on the different datasets.

The results indicate that a Gradient boosting algorithm is best suited for the prediction task, showing the best performance results, reasonable latency, offering comprehensibility and a high robustness. Regarding the different data types, models trained on sequential session data outperformed models trained on the static customer data by far.

Earlier research on predicting customer transactions have mainly been focused on using mathematical models, probabilistic algorithms and other mechanisms to determine the customer relations and establish claims which may not fit in every situation. Our approach to solving this puzzle is based on principles of artificial intelligence and proposes a machine learning model which can be used to predict customer purchase intent even when built on an anonymized dataset.

1.2 MOTIVATION

There is constant competition in the financial sector to provide better services and user experience than their rivals. One of the most valuable datasets which can be used is at the customer level of granularity. The increasing importance of classifying customer behavior and a large number of possible prediction models and data sources, this thesis aims at implementing and comparing suitable models trained on different datasets to identify the most appropriate one for predicting the abortion probability of a web shop visitor. Hence, a binary classification task of a visitor belonging to the aborting or not aborting category. It is, therefore, a suitable and relevant use case for testing classification models and different data types in the context of e-commerce.

Predicting customer behavior in the context of e-commerce is becoming more important nowadays. It increases customer satisfaction and sales, by facilitating an increase of customer experience through personalization, recommendations and special offers. Due to today's transition from visiting physical stores to online shopping, predicting customer behavior in the context of e-commerce is gaining importance. By utilizing clickstream and additional customer data, predictions can be carried out, ranging from customer classification, purchase prediction, and recommender systems to the detection of customer purchase intent.

A variety of machine learning models and data are available to conduct these kinds of predictions. Motivated by the increasing importance of classifying customer behavior and a large number of possible prediction models and data sources, this thesis aims at implementing and comparing suitable models trained on different datasets to identify the most appropriate one for predicting the abortion probability of a web shop visitor. Hence, a binary classification task of a visitor belonging to the aborting or not aborting category.

Customer transactions are one such source which could be about anything from opening a checking account to borrowing loans, purchasing a credit card etc. These actions along with personal details of the customer, the services used, ways in which they interacted with the bank, i.e., online, phone call or a physical visit, etc. help in characterization of customers further leading to implementing targeted strategies and forms one of the bases of customer segmentation. Web shop visitors leave more traces than ever before. Large amounts of personal information as well as clickstream data, recorded during each web shop visit, are collected, connected and stored for analysis with machine learning techniques.

1.3 CUSTOMER BEHAVIOUR ANALYSIS

Being a predictive analytics task, the aim of classification is to predict a categorical target variable from a set of input variables. This target variable can be expressed either through various categories or be of binary nature. The task at hand is a binary classification task since the target variable has two categories: buying and no buying. In order to predict the target variable a generalized relationship between input and target variable is learned from a labeled dataset. It is then applied to new data for classification.

Learning algorithms should both fit the training data and generalize well over new data. Various machine learning algorithms exist that have different methods of extracting this relationship. The most common type of machine learning algorithms for a binary classification task are vector-based methods. Belonging to this category are Logistic Regression, Random Forest, SVMs, KNN, and Gradient Boosting. The baseline model of this study also associates this category with being part of the Random Forest algorithms. They are eager learning models, where a classification model is constructed based on a given training dataset before new data is classified according to the model.

The opposites are lazy learners, such as the K-nearest Neighbor (KNN) algorithm, where training data is simply stored and a test data point is awaited for classification. All of the above-mentioned methods are stateless machine learning algorithms; they do not have any memory and always return the same answer given the same input. This is well-suited for most classification tasks, however, it is difficult to model patterns over time, since the different states have to be modeled through complex feature engineering, creating inaccuracies and increasing complexity by raising the number of input features.

Nevertheless, the ability to model time sequences and extract patterns over time can be useful for the research at hand, since the clickstream data used in this study is of sequential and time-dependent nature. Fortunately, stateful models exist, equipped with a memory, to remember previous states and to extract sequential patterns without the need for engineering time-dependent features explicitly. Resulting from the facts stated in this thesis, machine learning models are identified and implemented for solving the task of classifying a web shop visitor as aborting or non aborting, in the following referred to as no buying and buying sessions.

The models are applied to different data types, namely clickstream data generated by each visitor of the web shop as well as customer data if a visitor could be identified. This is done to establish which model and data is best-suited for the task of predicting the buying probability of an online shopping session, in terms of performance, latency, and comprehensibility. Dataset is acquired from UCI Repositories.

To provide further insight and reasoning about the varying performances on different datasets, an exploratory data analysis on the clickstream and static customer data is conducted. The features are selection and various binary classification machine learning models are trained to obtain the accuracy, precision, recall and AUC scores of those individual models. It is then to be further boosted to allow for a better accuracy.

CHAPTER 2

LITERATURE REVIEW

Jin-A Choi et al [1], there are several ongoing researches which try to classify the customers based on loyalty or retention , churn rate and interaction outcome analysis . Most of these researches do not explicitly address predicting customer transactions, but instead it is treated as a derived outcome of their research. One of the many problems with the predictive modelling techniques used in such researches is high dimensionality. This can be handled by using Principle Component Analysis, variable importance from Random Forests and several other techniques. But these methods find the important features with a certain trade-off of variance which can be put down as the initial assumption.

Devendra Prakash Jaiswal et al [2], proposed to classify the visitor behaviour patterns with the aim of determining the Web site component that has the highest impact on the fulfilment of business objectives. Factors from PCA can explain most of the variance present in the data and identify the important features with an assumption that data is linearly separable , the assurance of which cannot be guaranteed because of its anonymity, i.e., if inconsistent anonymization algorithms are used for encoding features, the data might lose its statistical integrity. Other approaches like variable importance from Random Forest carry with itself a certain amount of bias which might hamper our predictions because we know nothing about the predictor features.

Pena-García et al [3], also showed that the navigation paths of visitors in the e-commerce site can be used to predict the actions of the visitors. In contrast to an analytical approach, a Multi-Agent Based Simulation (MABS) does not try to

model the system in its entirety but splits it up into interacting entities (called agents), whose behaviours are sometimes simpler to implement rather than finding equations describing the global system. The core idea of MABS relies on the concept of emergence: the overall dynamic of the system results from their individual behaviours and their interactions. In addition, we can note that agent-oriented approaches have been used, for a long time, for trading in virtual markets where agents could autonomously purchase or sell products.

Arnaud Doniec et al [4], in another study, considering the loss of throughput in Web Servers due to overloading, proposed a system to assign priorities to sessions according to the revenue that will be generated using early clickstream data and session information. The training dataset was created from 7000 transactions, of which half were chosen to be from ‘‘buying’’ class to deal with class imbalance problems. They used Markov chains, logistic linear regression, decision trees, and Naive Bayes to generate a probability on the customer purchasing intention.

Alireza Ghahtarani et al [5], on another major challenge is the anonymity of the features and values blurring our understanding of the data which is the first step of solving any problem. Apart from the fact that the structure of the data is preserved, we cannot claim anything about its statistical properties since the anonymization of data is done using techniques that are unknown to us. For us, this principle fails as due to high dimensionality, fixating a threshold for feature identification might lead us astray in our analysis especially when the exploration provides no valuable insights due to data anonymity. Therefore, we propose a methodology of predicting customer transactions using the conventional machine learning techniques on a high dimensional anonymized data and trying to address this problem directly instead of deriving it from some other consequences .

CHAPTER 3

PROPOSED WORK

3.1 DATASET DESCRIPTION

In this part, the data being used for the training and testing of the models is described. The dataset has been obtained from UCI repositories. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. It is used by students, educators, and researchers all over the world as a primary source of machine learning data sets. As an indication of the impact of the archive, it has been cited over 1000 times.

The dataset consists of feature vectors belonging to 12,330 sessions. The dataset we used has been obtained from University of California Irvine repositories. The features that exist in the dataset are Admin, Ad. Duration, Info, Info Duration, Prod, Prod duration, Bounce Rate, Exit rate, Page Value, Special day, OperatingSystem, Browser, Region, TrafficType, VisitorType, Weekend, Month, Revenue. Out of the 12,330 sessions we have analysed that 84.5%(10,422) were negative class samples where the customer purchase intent did not end with shopping. The rest 15.5%(1908) resulted in positive class samples ending with shopping.

Feature name	Feature description	Number of Values
OperatingSystems	Operating system of the visitor	8
Browser	Browser of the visitor	13
Region	Geographic region from which the session has been started by the visitor	9
TrafficType	Traffic source (e.g., banner, SMS, direct)	20
VisitorType	Visitor type as "New Visitor," "Returning Visitor," and "Other"	3
Weekend	Boolean value indicating whether the date of the visit is weekend	2
Month	Month value of the visit date	12
Revenue	Class label: whether the visit has been finalized with a transaction	2

Figure 3.1 Numerical Features

Feature name	Feature description	Min. val	Max. val	SD
Admin.	#pages visited by the visitor about account management	0	27	3.32
Ad. duration	#seconds spent by the visitor on account management related pages	0	3398	176.70
Info.	#informational pages visited by the visitor	0	24	1.26
Info. durat.	#seconds spent by the visitor on informational pages	0	2549	140.64
Prod.	#pages visited by visitor about product related pages	0	705	44.45
Prod.durat.	#seconds spent by the visitor on product related pages	0	63,973	1912.3
Bounce rate	Average bounce rate value of the pages visited by the visitor	0	0.2	0.04
Exit rate	Average exit rate value of the pages visited by the visitor	0	0.2	0.05
Page value	Average page value of the pages visited by the visitor	0	361	18.55
Special day	Closeness of the site visiting time to a special day	0	1.0	0.19

Figure 3.2 Categorical Feature

3.2 PROPOSED ARCHITECTURE

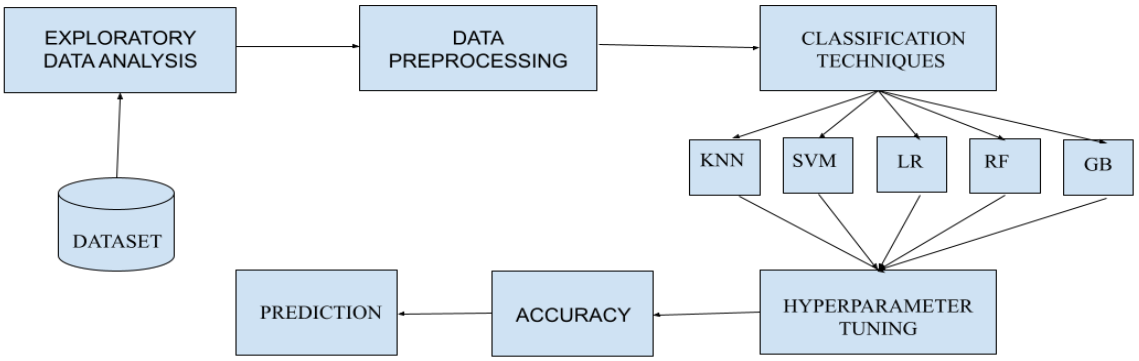

Figure 3.3 Architecture

Our aim is to study and analyse the behaviour of customers in order to find out whether they will be interested in a purchase intent or not. The customer behavior analysis is performed on e-commerce websites for an in depth study regarding the purchase intent of a customer using their online behaviour. The dataset required is obtained from University of California Irvine repositories - Center for Machine Learning and Intelligent Systems. It consists of 10 numerical and 8 categorical attributes.Initially we will be performing statistical analysis on our dataset followed by data cleaning, this will eliminate the data missing points and values.After data cleaning, data type fix is performed in which boolean features are changed to binary so that it can be used for further calculations.

Furthermore Exploratory Data Analysis (EDA) will be performed in which correlation analysis, web page analysis, page metric analysis, visitor analysis, visitor data analysis is implemented to obtain a deeper understanding of dataset and customer behaviour.In data pre-processing, we will make our data ready for

model training. This includes data transformation, data splitting and data cleaning. Data splitting is performed to split the data for training and testing.

Furthermore, feature Scaling is one of the important preprocessing that is required for standardizing/normalization of the input data. We will scale the features in our subsets, in order to use them to train, validate, and test models that will benefit from data scaling. The next step taken is model building.

We have shortlisted the following machine models, Support Vector Machine(SVM), Random Forest Technique and Gradient Boosting.To find the efficiency of the above models, we will be performing the accuracy, F1 score, precision and recall.Furthermore, to improve the efficiency we will be performing tuning. This will improve the performance of the model.The performance of every model will be compared and an ROC (receiver operating characteristic) curve will be generated. Using the results obtained we will be able to find the customer purchase intent using behavioural analysis.

3.3 EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) is used by data scientists to analyse and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily

EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed.

It refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

EDA helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. In particular, confusing the two types of analyses and employing them on the same set of data can lead to systematic bias owing to the issues inherent in testing hypotheses suggested by the data. The objectives of EDA are to:

- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments.

Many EDA techniques have been adopted into data mining. They are also being taught to young students as a way to introduce them to statistical thinking. EDA is primarily used to see what data can reveal beyond the formal modelling or hypothesis testing task and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

3.4 DATA PRE-PROCESSING

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, we use data pre-processing tasks.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Data preprocessing is an important step in the machine learning process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations and missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running any analysis. Often, data preprocessing is the most

important phase of a machine learning project, especially in computational biology.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take a considerable amount of processing time. Data preprocessing includes cleaning, Instance selection, normalization, transformation, feature extraction and selection, etc. The product of data preprocessing is the final training set.

Data preprocessing may affect the way in which outcomes of the final data processing can be interpreted. This aspect should be carefully considered when interpretation of the results is a key point. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way.

So, for this, we use data pre-processing tasks. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

3.4.1 Data Transformation

Data transformation is the process of converting data from one format to another, typically from the format of a source system into the required format of a destination system. Data transformation is a component of most data integration and data management tasks, such as data wrangling and data warehousing. The goal of the data transformation process is to extract data from a source, convert it into a usable format, and deliver it to a destination.

Both artificial intelligence and machine learning business use cases need vast amounts of data to train the algorithms. For the most accurate results – the ones that you want to base insights-driven decisions on – that data needs to be in an analytics-ready state. The data should be joined together, of the highest quality, and embellished with appropriate metrics that the algorithms can use. Simply put, data transformation makes your data useful.

Data transformation is the process in which you take data from its raw, siloed and normalized source state and transform it into data that's joined together, dimensionally modeled, denormalized, and ready for analysis. Without the right technology stack in place, data transformation can be time-consuming, expensive, and tedious. Nevertheless, transforming your data will ensure maximum data quality which is imperative to gaining accurate analysis, leading to valuable insights that will eventually empower data-driven decisions.

Building and training models to process data is a brilliant concept, and more enterprises have adopted, or plan to deploy, machine learning to handle many practical applications. But for models to learn from data to make valuable predictions, the data itself must be organized to ensure its analysis yield valuable insights.

When it comes to machine learning, you need to feed your models good data to get great insights, and in most cases, some sort of data cleansing needs to be performed prior to any data analysis. This is a critical step as it ensures data quality, which increases the accuracy of predictions. Data transformation is a component of most data integration and data management tasks, such as data wrangling and data warehousing. The goal of the data transformation process is to extract data from a source, convert it into a usable format, and deliver it to a destination.

3.4.2 Feature Selection

The feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset. It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion.

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance.

Feature selection and Data cleaning should be the first and most important step of your model designing. Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.

Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features. It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion.

Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for several reasons:

- simplification of models to make them easier to interpret by researchers/users,
- shorter training times,
- to avoid the curse of dimensionality,
- enhanced generalization by reducing overfitting.

3.4.3 Data Scaling

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization.

For example, many classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it

It's also important to apply feature scaling if regularization is used as part of the loss function (so that coefficients are penalized appropriately). It is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

3.5 MODEL SELECTION AND BUILDING

A common problem in Machine Learning is model selection, which means determining which model performs the best with your data. You need to select the best model across different class of algorithms, like Random Forest or K-NN, and different sets of hyperparameter like the Learning Rate or the Number of Iterations. Indeed picking the best algorithm is not enough, each algorithm make use of different parameters that need to be adjusted to produce the best performing model. Various machine learning algorithms that are chosen for training model are as follows:

- K-Nearest Neighbor
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- Gradient Boosting

3.6 MODEL EVALUATION AND METRICS

3.6.1 Test Train Split

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluatemodel performance.

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

In this project, we have validated using Cross-Validation method for evaluation. In this method, the mostly large dataset is randomly divided into two subsets:

1. Training set
2. Test set

3.6.2 Metrics

Confusion Matrix :

A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data. The matrix is NxN, where N is the number of target values (classes). Performance of such models are commonly evaluated using the data in the matrix.

Various measures exist to assess and compare the performance of machine learning models on a binary classification task. These metrics are based on the so-called confusion matrix, from which one can derive the correctly predicted cases, indicated in green, called true positives and true negatives. These are the cases where a visitor did not purchase anything and a no buying session was predicted and sessions where a purchase occurred and was also predicted.

Also, the wrongly predicted cases can be identified, as indicated in orange, the false negatives, and the false positives, where a purchase occurred but none was predicted or where no purchase occurred but one was predicted. From the confusion matrix, various performance metrics can be derived. All metrics calculated from the confusion matrix depend on the chosen classification threshold, based on which the confusion matrix was created.

Classification Accuracy :

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of the number of correct predictions to the total number of input samples. the proportion of the total number of predictions that were correct. It works well only if there are an equal number of samples belonging to each

class. Model accuracy is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. It's a way of assessing the performance of a model.

The proportion of the total number of predictions that were correct. Model accuracy is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. It's a way of assessing the performance of a model.

Positive Predictive Value or Precision :

The proportion of positive cases that were correctly identified. In pattern recognition, information retrieval and classification, precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that were retrieved.

In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labelled as belonging to the positive class) divided by the total number of elements labelled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labelled as belonging to the class).

Recall score :

The ability of the classifier to find all the positive samples. Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labelled as belonging to the positive class but should have been).

The precise definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. Recall can be thought of as a model's ability to find all the data points of interest in a dataset.

Area Under Curve Score:

The Area Under Curve(AUC) represents the probability that a random positive example is positioned to the right of a random negative example. It is the area under the ROC (Receiver Operating Characteristic) curve in percentage. Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problems. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. Before defining AUC, let us understand basic terms :

- True Positive Rate (Sensitivity) : True Positive Rate is defined as TP/ (FN+TP). True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.

- True Negative Rate (Specificity) : True Negative Rate is defined as TN / (FP+TN). False Positive Rate corresponds to the proportion of negative data points that are correctly considered as negative, with respect to all negative data points.

- False Positive Rate : False Positive Rate is defined as FP / (FP+TN). False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

PROCESSOR : Intel Core i3 8th Gen

HARD DISK DRIVE : 1TB

RAM : 8 GB

4.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM : Microsoft Windows 10

PROGRAMMING LANGUAGES : Python 3

IDE : Visual Studio Code

4.3 DESCRIPTION OF THE TOOLS USED

The project is mainly implemented in Python. The following are the tools used:

- Python
- Visual Studio Code
- Pandas
- Seaborn

- Numpy
- Matplotlib

4.3.1 Python

Python web server, or Jupyter document format depending on context. Python is an interpreted, high-level, general-purpose programming language. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management.[66] It also features dynamic name resolution (late binding), which binds method and variable names during program execution. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, it also has a comprehensive standard library.

4.3.2 Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS and Linux. It is a freeware source-code editor made by Microsoft. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages such as C++, C#, Java, Python, PHP, Go.

4.3.3 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

The library features include Data Frame objects for data manipulation with integrated indexing, tools for reading and writing data between in- memory data structures and different file formats, data alignment and integrated handling of missing data, reshaping and pivoting of data sets etc.

4.3.4 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across

platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits. It provides an object- oriented API for embedding plots into applications using general-purposeGUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB.

4.3.5 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them. It provides a high-level interface for drawing attractive and informative statistical graphics.

4.3.6 Numpy

NumPy stands for ‘Numerical Python’. It is an open-source Python library used to perform various mathematical and scientific tasks. It contains multi-dimensional arrays and matrices, along with many high-level mathematical functions. Many of its functions are very useful for performing any mathematical or scientific calculation. As it is known that mathematics is the foundation of machine learning, most of the mathematical tasks can be performed using NumPy.

4.3.7 Scikit-learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

In such cases, extending these methods with Python may not be possible. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

CHAPTER 5

IMPLEMENTATION MODULES

5.1 EXPLORATORY DATA ANALYSIS

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, and find interesting relations among the variables. Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals.

Exploratory Data Analysis can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning.

Exploratory Data Analysis is a complement to inferential statistics where it is mixed with rigid rules and formulas whereas, Data Analysis is the combination of statistics and probability to figure out the trends and patterns of the dataset. EDA is the first step in the Data Analysis phase where you can manipulate the datasets to achieve the results.

It is implemented before the statistical techniques are applied to the datasets. Statistical techniques are usually applied on the datasets with the histogram or box plots but EDA does not come with a set of techniques or

procedures. It is more like a form of art than applying science. The EDA process makes the analyst get a feel about the dataset and uses their ideas to judge the important elements in the dataset.

For example, in multidimensional scaling, it is the visual representation of the distance of similarities between the set of objects. The user can find the exact distance between the objects by looking at the multidimensional representation. The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, and find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning. EDA is a crucial step before getting deep into machine learning or modeling your data to solve your business problems. It allows you to analyze the proper model to interpret the correct results.

Machine learning has more powerful advanced algorithms and so, people almost skip the Exploratory Data Analysis phase. People usually take advantage of algorithms and skip the EDA phase, where it is like feeding the data into the black box and look for better results. Exploratory Data analysis provides a lot of crucial information where people usually miss and this information helps in the long run.

5.1.1 Correlation Analysis

Correlation explains how one or more variables are related to each other. These variables can be input data features which have been used to forecast our target variable. Correlation, statistical technique which determines how one variable moves/changes in relation with the other variable.

It gives us the idea about the degree of the relationship of the two variables. If two variables are closely correlated, then we can predict one variable from the other.Correlation plays a vital role in locating the important variables on which other variables depend.It's used as the foundation for various modeling techniques.Proper correlation analysis leads to better understanding of data.It's a bi-variate analysis measure which describes the association between different variables. In most of the business it's useful to express one subject in terms of its relationship with others.

Positive Correlation is two features (variables) can be positively correlated with each other. It means that when the value of one variable increase then the value of the other variable(s) also increases.

Negative Correlation of two features (variables) can be negatively correlated with each other. It means that when the value of one variable increases then the value of the other variable(s) decreases.

No Correlation are two features (variables) are not correlated with each other. It means that when the value of one variable increases or decreases then the value of the other variable(s) doesn't increase or decrease.

Inference

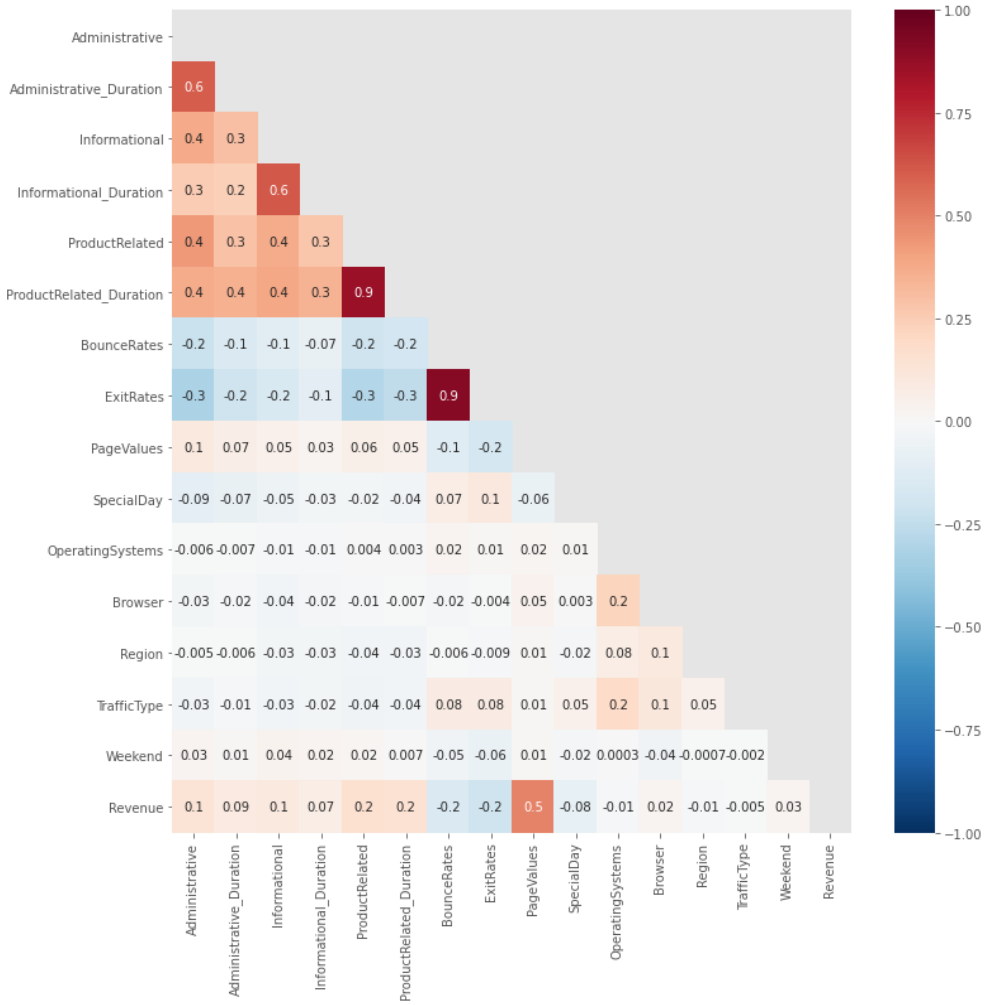

Figure 5.1 Correlation Analysis

From the above heatmap, we observe there is very little correlation among the different features in our dataset, the very few cases of high correlation (|corr| >= 0.7) are bounce rates and exit rates (0.9), product related and product related duration (0.9), moderate correlations (0.3 < |corr| < 0.7). Among the following features - Administrative, Administrative_Duration, Informational, Informational_Duration, ProductRelated, and ProductRelated_Duration, also between PageValues and Revenues.

5.1.2 WEB PAGE ANALYSIS

Web page analysis in Exploratory Data Analysis is the process of analysing the dataset which includes the customer data in regards to their activity on the website. By analysing the actions that the customer makes on the website, we can likely predict certain actions that are to be made by the customer or at least understand the general action and study the customer behaviour.

Inference

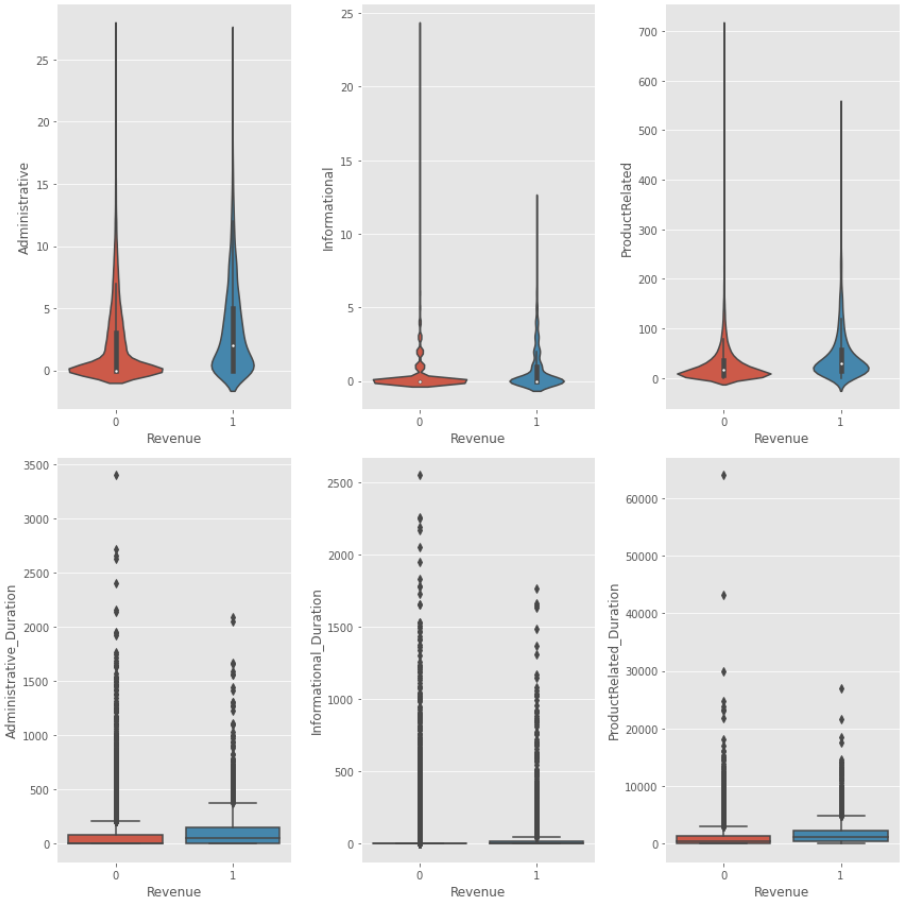

Figure 5.2 Web Page Analysis

From the above boxplots, we can see that in general, visitors tend to visit less pages, and spend less time, if they are not going to make a purchase. The number of product related pages, and the time spent on them, is way higher than that for account related or informational pages. The first 3 features look like they follow a skewed normal distribution.

5.1.3 PAGE METRIC ANALYSIS

Website metrics consist of data that can be analysed for feature importance and relation which is crucial in feature selection and increasing the accuracy and the efficiency of target models.

Inference

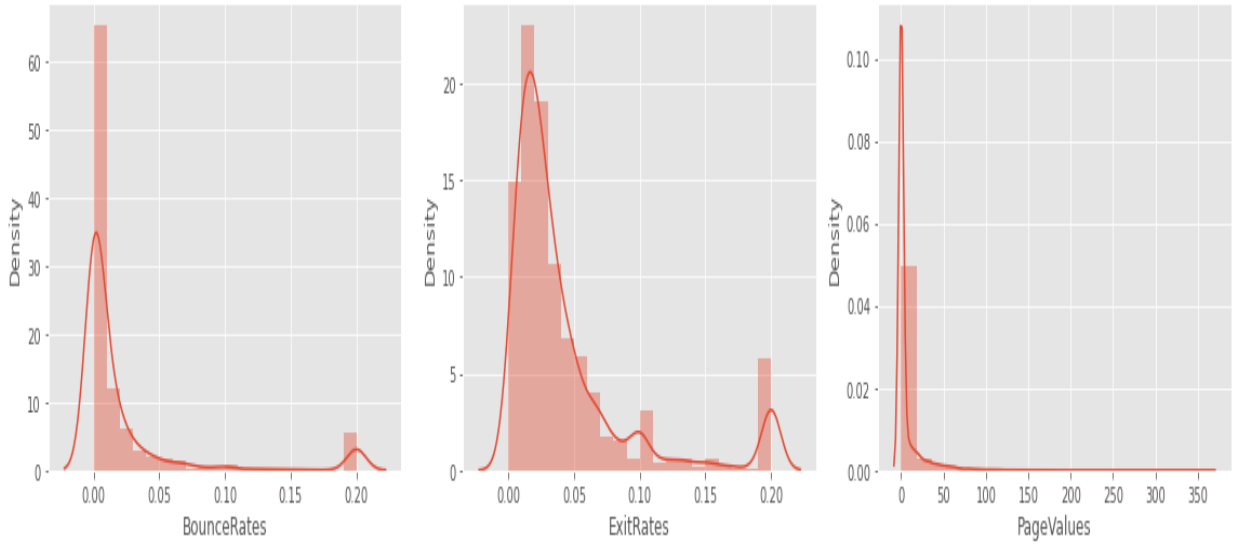

Figure 5.3 Page Metric Analysis

From the above visualizations of 3 google analytics metrics, we can conclude BounceRates & PageValues do not follow a normal distribution. All 3 features have distributions that are skewed right. All 3 distributions have a lot of outliers. The average bounce and exit rates of most of our data points is low,

which is good, since high rates identicate that visitors are not engaging with the website. Exit rate has more high values than bounce rate, which makes sense, where transaction confirmation pages for example will cause the average exit rate to increase. Bounce rate is the percentage where the first page visited was the only page visited in that session. Exit rate of a page is the percentage where that page was the last page visited in the session, out of all visits to that page.

5.1.4 VISITOR ANALYSIS

Visitor Analysis is a form of EDA that involves studying the visitor data that has been collected. It can be very helpful in identifying key factors in visitor data. It provides data such as region, traffic type, operating system and browser to provide analysis of visitor data

Inference

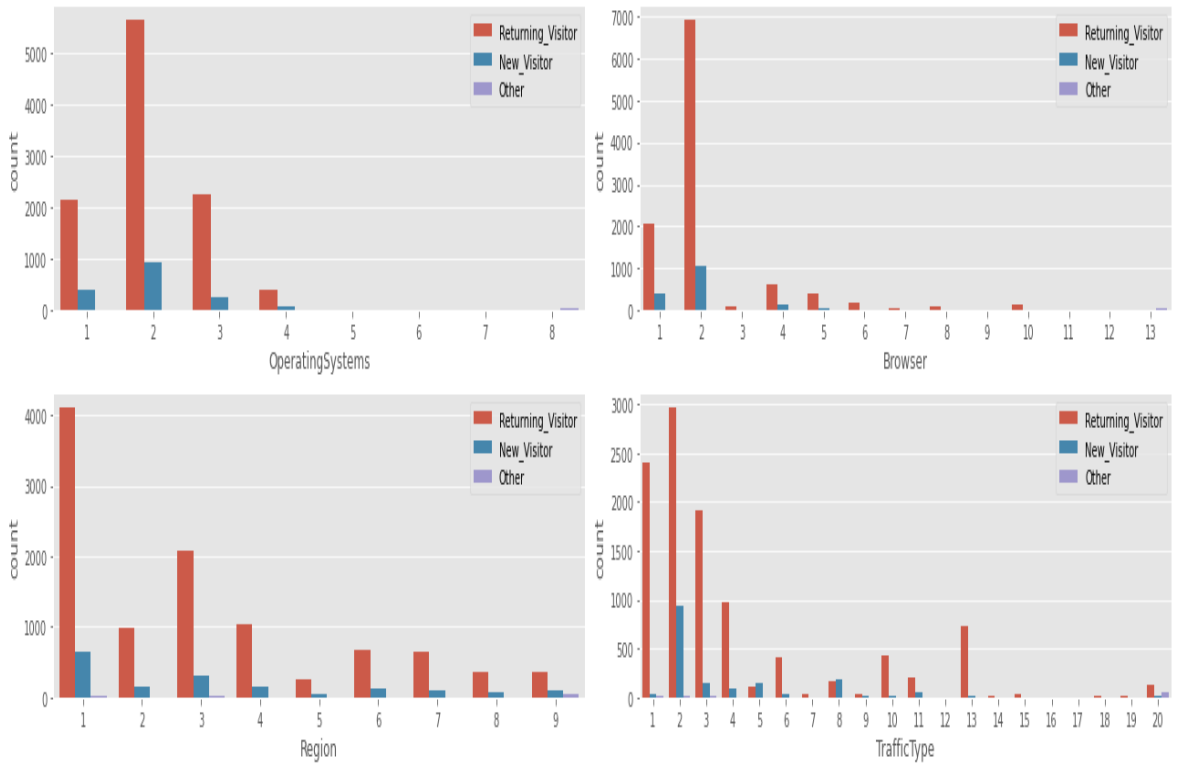

Figure 5.4 Visitor Analysis

Operating system is responsible for ~7000 of the examples in our dataset. 4 of the 8 operating systems used, are responsible for a very small number (<200) of the examples in our dataset. A similar story repeated with the browsers used by visitors, where there is 1 dominant browser, 3 with decent representation in the dataset, and the rest are rarely used. It looks like we have a very regionally diverse traffic in our dataset. Also Traffic sources are very diverse, with a few that did not contribute much to the dataset.

5.1.5 VISITOR DATE ANALYSIS

Visitor date analysis is another form of EDA where the calendrical data of the customer is analysed to understand the key factor in terms of calendar. It can help understand patterns and its driving forces. Calendrical data of the customer is analysed to understand the key factor in terms of calendar in EDA.

Inference

In March and May, we have a lot of visits (May is the month with the highest number of visits), yet transactions made during those 2 months are not on the same level. We have no visits at all during January nor april. Most transactions happen during the end of the year, with Nov as the month with the highest number of confirmed transactions. The closer the visit date to a special day (like black Friday, new year's, ... etc) the more likely it will end up in a transaction. Most transactions happen on special days (SpecialDay =0).

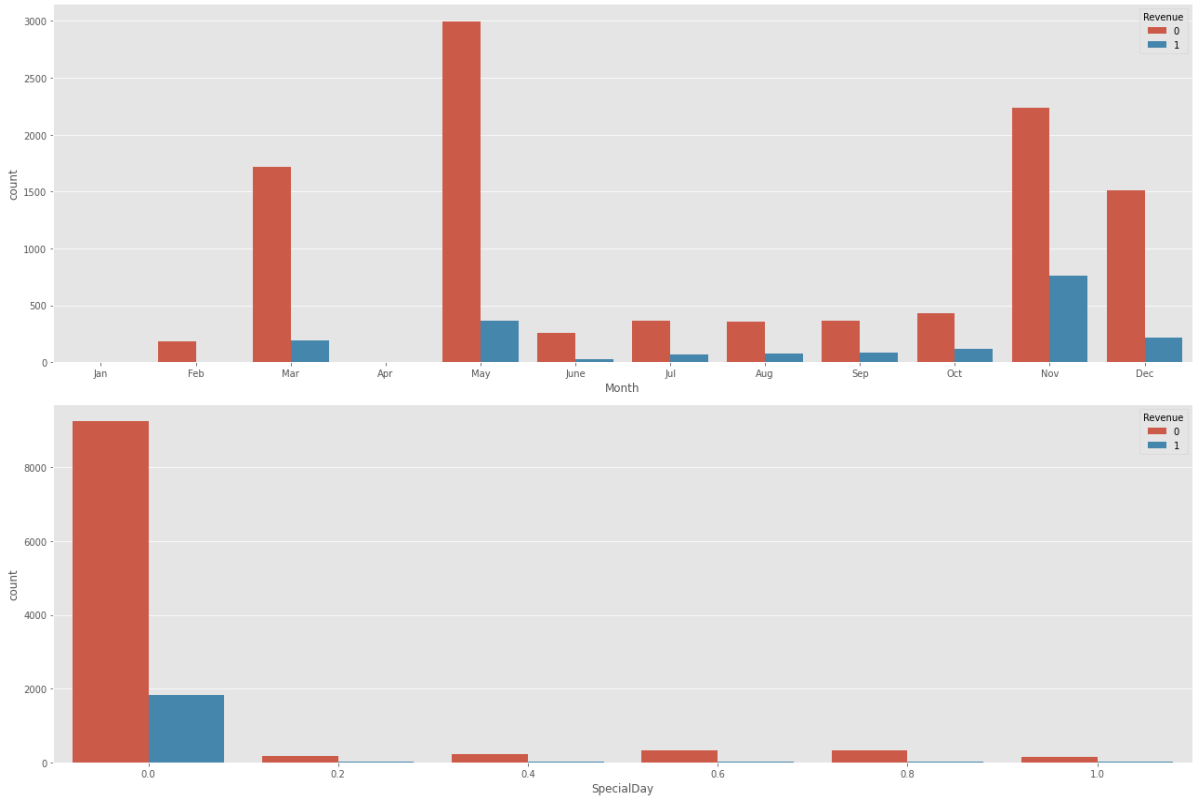

Figure 5.5 Visitor Date Analysis

5.2 K-NEAREST NEIGHBOUR

K-NN is a supervised learning classification algorithm. K-NN algorithm predicts the class label of a new input. K-NN utilizes the similarity of new input to its inputs samples in the training set. If the new input is the same as the samples in the training set, the K-NN classification performance is not good.

The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood calculating the distance between points on a graph. K-NN utilizes the similarity of new input to its input samples in the training set.

As mentioned before, the KNN is a lazy learning algorithm, where training data is simply stored and a test data point is awaited for classification. All of the stored training instances correspond to points in an n-dimensional feature space. A point's nearest neighbors are defined by distance measurements, most commonly Euclidean distance.

An unlabeled test data point will be assigned the label most common amongst its k-nearest neighbors. The advantages of this method are its robustness to noisy data and a very fast training speed. Disadvantages are an increased complexity of dimensionality through irrelevant features and therefore a decreased performance, highlighting the importance of feature engineering, longer prediction times compared to eager learning models and a low comprehensibility with high-dimensional input.

K-NN utilizes the similarity of new input to its input samples in the training set. If the new input is the same as the samples in the training set, the K-NN classification performance is not good. The KNN model has been applied regularly in e-commerce regarding recommender systems, where products are recommended to a web shop visitor based on the preferences of its nearest neighbors.

The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood calculating the distance between points on a graph.

Inference

The performance of the K-NN model is inferred whose measures are accuracy score = 87.3%, f1 score = 50.2%, precision score = 72.8%, recall score = 38.3% and AUC ROC curve = 68%.

KNN's initial model provides accuracy and F1 score, with a decrease in recall. Now we will try to increase its performance even more with hyper-parameters in KNN tuning via grid search. The obtained results after tuning are accuracy score = 87.3%, f1 score = 50.6%, precision score = 72.2%, recall score = 38.9% and AUC ROC curve = 68%.

5.3 SUPPORT VECTOR MACHINE

SVM is a machine learning algorithm used for classification/ regression .It classifies both linear and non-linear data. It separates data based on labels. The technique, kernel trick used to match new data to best from training data to predict unknown target labels. The SVM finds this hyperplane using support vectors and margins. SVM performs classification tasks by maximizing the margin separating both classes while minimizing the classification errors.

SVM is a machine learning algorithm used for classification/ regression. An SVM separates two classes by fitting a hyperplane between them. Doing this, only one hyperplane is used, which differs from Random Forest where a hyperplane is added after each split. In cases where multiple separating hyperplanes can be found, the SVM detects the maximum-margin hyperplane, maximizing the distance to data points of both classes. This leads to a higher generalizability and therefore to better test accuracies. It classifies both linear and non-linear data. It separates data based on labels.

The technique, kernel trick used to match new data to best from training data to predict unknown target labels. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups. It applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

The SVM finds this hyperplane using support vectors and margins. SVM performs classification tasks by maximizing the margin separating both classes while minimizing the classification errors. If a feature space is not linearly separable, a kernel-function is used to map data on a higher dimensional feature space where the data becomes linear separable.

Regarding their performance, SVMs show a high accuracy as well as fast prediction times. They further work very well with high-dimensional data input. Disadvantages are long training times and their difficulty to be interpreted. Further, next to feature engineering, also hyper-parameter tuning is required, which can be difficult and time-consuming

Inference

The performance of the support vector machine model is inferred whose measures are accuracy score = 88.7%, f1 score = 58.5%, precision score = 75.7%, recall score = 47.7% and AUC ROC curve = 73%.

SVM's initial model resulted in a considerable increase in all performance metrics. Now we will try to increase its performance even more with hyper-parameter tuning via grid search.The obtained results after tuning are accuracy score = 88.9%, f1 score = 60%, precision score = 75.1%, recall score = 50%.

5.4 LOGISTIC REGRESSION

Logistic Regression is a classification algorithm mostly used for binary classification problems. In logistic regression instead of fitting a straight line or hyper plane, the logistic regression algorithm uses the logistic function to squeeze the output of a linear equation between 0 and 1. There are 12 independent variables which makes logistic regression good for classification.

Logistic regression belongs to the class of generalized linear models and is used to predict categorical target variables. This is achieved through a logistic function, which has the shape of a sigmoid curve, taking values between 0 and 1. In a binary logistic regression model, the dependent variable has two levels (categorical).

Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression (for example the proportional odds ordinal logistic model). The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier. The coefficients are generally not computed by a closed-form expression, unlike linear least squares.

This function is modeled by combining input values linearly with coefficients. Where y is the output, b0 the bias term and b1 the coefficient for the input value x. In logistic regression instead of fitting a straight line or hyper plane, the logistic regression algorithm uses the logistic function to squeeze the output of a linear equation between 0 and 1.

There are 12 independent variables which makes logistic regression good for classification. Every column of the input vector learns a coefficient from the training data through maximum-likelihood estimation. LR is very fast regarding prediction and training times, it is, hence, one of the most popular machine learning algorithms for binary classification. It, nevertheless, requires feature engineering and the encoding of categorical variables.

It is also sensitive to noise, therefore, outliers should be removed before the training. Further, LR does not perform well on highly correlated input factors. Hence, it can be helpful to only use principal components for the regression. This can be achieved through a Exploratory Data Analysis (EDA), a method to extract the principal components from a set of possibly correlated variables.

Inference

The performance of logistic regression model is inferred whose measures are accuracy score = 87.7%, f1 score = 52.3%, precision score = 74.6%, recall score = 40.2% and AUC ROC curve = 69%.

The logistic regression classifier resulted in less accuracy and F1 score compared to SVM. Tuning it hyper-parameters to achieve better performance.The obtained results after tuning are accuracy score = 87.9%, f1 score = 52.8%, precision score = 75.7%, recall score = 40.5%.

5.5 RANDOM FOREST

Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes predictions based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing. The generated samples from the decision tree can be saved so that it can be used on other data. In the random forest there are two stages, firstly create a random forest then make a prediction using a random forest.

Bagging is another example for ensemble trees, where many large trees fit to the bootstrap re-sampled versions of the data and are classified by majority vote. Random Forest improves on Bagging by de-correlating the trees. After each tree split a random sample of features is chosen and only these are considered for the next split.

The results are again based on the majority vote of the single trees. By using a large number of classifiers, Decision trees are a popular method for various machine learning tasks.

Tree learning comes closest to meeting the requirements for serving as an off-the-shelf procedure for data mining because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate".

In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.

This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model. Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a K-fold cross validation.

Bagging and Random Forest, improve on the weaknesses of non-ensemble K-NN, such as robustness and over-fitting. They train faster than the boosted trees but need more time for the prediction. Nevertheless, Bagging and Random Forest still depend on feature engineering and cannot model time dependencies. The generated samples from the decision tree can be saved so that it can be used on other data. In the random forest there are two stages, firstly create a random forest then make a prediction using a random forest.

Inference

The performance of the random forest model is inferred whose measures are accuracy score = 89.4%, f1 score = 62.2%, precision score = 77%, recall score = 52.% and AUC ROC curve = 77%.

The random forest classifier (with default parameter values) gives us higher accuracy and F1 score than all other classifiers tested so far. Next, we will try improving its performance (especially the recall) by tuning its hyper-parameters. The obtained results after tuning are accuracy score = 90.1%, f1 score = 66.%, precision score = 77.3%, recall score = 57.7%.

5.6 GRADIENT BOOSTING

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

In boosting, each new tree is a fit on a modified version of the original data set. The gradient boosting algorithm (gbm) can be most easily explained by first introducing the AdaBoost Algorithm.The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight. After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify.

The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree. Our new model is therefore Tree 1 + Tree 2. We then compute the classification error from this new 2-tree ensemble model and grow a third tree to predict the revised residuals.

We repeat this process for a specified number of iterations. Subsequent trees help us to classify observations that are not well classified by the previous trees. Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models. Gradient Boosting trains many

models in a gradual, additive and sequential manner. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest.

It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Inference

The performance of the gradient boosting machine model is inferred whose measures are accuracy score = 90.5%, f1 score = 68.9%, precision score = 76%, recall score = 62.9% and AUC ROC curve = 80%.

The default classification performance of gradient boosting is better than that of random forest. Next, to improve its performance even more by tuning its hyper-parameters. The obtained results after tuning are accuracy score = 90.6%, f1 score = 69.5%, precision score = 76.5%, recall score = 63%.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 VISUALIZATION OF DATA

Data visualization is the representation of data or information in a graph, chart, or other visual format. It communicates relationships of the data with images. This is important because it allows trends and patterns to be more easily seen. With the rise of big data upon us, we need to be able to interpret increasingly larger batches of data.

Machine learning makes it easier to conduct analyses such as predictive analysis, which can then serve as helpful visualizations to present. But data visualization is not only important for data scientists and data analysts, it is necessary to understand data visualization in any career. Whether you work in finance, marketing, tech, design, or anything else, you need to visualize data. That fact showcases the importance of data visualization.

We need data visualization because a visual summary of information makes it easier to identify patterns and trends than looking through thousands of rows on a spreadsheet. It's the way the human brain works. Since the purpose of data analysis is to gain insights, data is much more valuable when it is visualized. Even if a data analyst can pull insights from data without visualization, it will be more difficult to communicate the meaning without visualization. Charts and graphs make communicating data findings easier even if you can identify the patterns without them.

Figure 6.1 Feature Relation Analysis

6.2 RESULT ANALYSIS

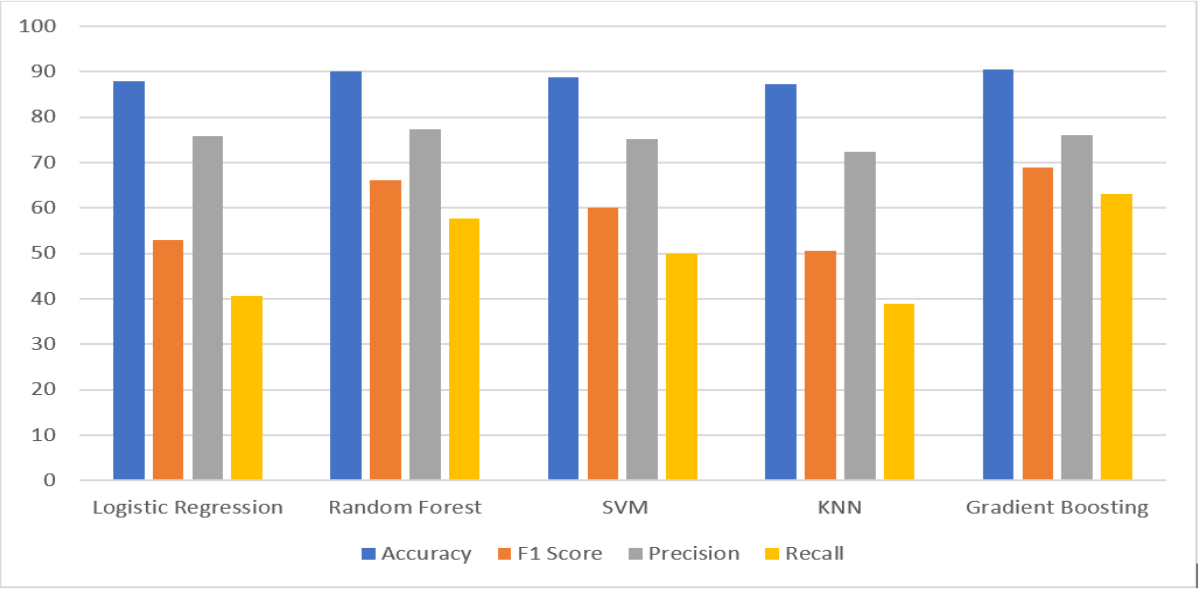

Figure 6.2 Comparison of models after tuning

The dataset consists of 12,330 sessions. The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period. Of the 12,330 sessions in the dataset, 84.5% (10,422) were negative class samples that did not end with shopping, and the rest (1908) were positive class samples ending with shopping. After preprocessing the data, data mining classification techniques such as Logistic Regression, Random Forest, Support Vector Machine, K-Nearest Neighbours, Artificial Neural Networks, Gradient Boosting were applied and their metrics are compared. The algorithms arranged in descending order of their accuracy after tuning are Gradient Boosting(90.5%), Random Forest(90.2%), KNN(87.3%), Logistic Regression(87.9%) and SVM(88.9%). From the above results we have received the highest accuracy with Gradient Boosting which delivered 90.5% accuracy.

CHAPTER 7

CONCLUSION AND FUTURE WORK

With the increase in the need for marketing analytics in modern day companies, it has become evident that a system that can analyse and predict customer purchase intent is very much required in the industry. The motivation of this study was to analyse customer behaviour and build a machine learning model that can accurately and efficiently predict the customer purchase intent. This study compares the accuracy score of Logistic Regression, Random Forest, Support Vector Machine, K- Nearest Neighbours and Gradient Boosting algorithms for predicting customer transactions using the dataset obtained from UCI repositories.

First, any follow-up study should be concerned with running the different algorithms on a more computationally powerful machine instead of locally on a regular desktop computer. All algorithms offer the possibility of parallelization, which can also be utilized on a better machine. This can already help to create more powerful models, through using more data, conducting a more thorough hyper-parameter tuning and running more training iterations to fully exploit and analyze the algorithms' potential. Regarding the data, it would be interesting if using other types of features, especially ones that regard aspects such as click information, results in better predictions. Therefore, a follow up study could focus on using more information of the data that is available anyways to see if performance can also be enhanced by improving the data basis instead of tuning and exploring the potentials of different algorithms.

REFERENCES

1. Choi, J. A., & Lim, K., (2020), “Identifying machine learning techniques for classification of target advertising”, ICT Express, Elsevier B.V. Vol. 27, pp. 23-31.

2. Jaiswal, D. P., Kumar, S., & Mukherjee, P., (2020), “Customer Transaction Prediction System”, Procedia Computer Science, Elsevier B.V, Vol.168, pp. 49-56.

3. Peña-García, N., Gil-Saura, I., Rodríguez-Orejuela, A., & Siqueira-Junior, J. R., (2020), “Purchase intention and purchase behavior online: A cross-cultural approach”, Heliyon, Vol.168, pp. 34-37.

4. Doniec, A., Lecoeuche, S., Mandiau, R., & Sylvain, A., (2020), “Purchase intention-based agent for customer behaviours”, Information Sciences, Vol.521, pp.380-397.

5. Ghahtarani, A., Sheikhmohammady, M., & Rostami, M., (2020), “The impact of social capital and social interaction on customers’ purchase intention, considering knowledge sharing in social commerce context”, Journal of Innovation & Knowledge, Vol.5(3), pp.191-199.