INTERNATIONAL HELLENIC UNIVERSITY

Predicting Customer Purchase Intention Using Online Machine Learning Methods

Eleftheria Trigeni

SID: 3308210043

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

APRIL 2023

THESSALONIKI – GREECE

Predicting Customer Purchase Intention Using Online Machine Learning Methods

Eleftheria Trigeni

SID: 3308210043

Supervisor:	Dr. D. Karapiperis
Supervising Committee Members:	Assoc. Prof. K. Diamantaras
	Assist. Dr. P. Koukaras

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

APRIL 2023

THESSALONIKI – GREECE

Abstract

E-commerce has gained significant popularity and this trend is expected to grow even more. The users do not always visit an e-commerce with purchase intention but with browsing one meaning that they interact with the products but ultimately abandon the website without offer revenue to the business. This is a challenge that any e-commerce faces. Users interacting with it generate a huge volume of data that business can benefit from to convert browsers into buyers and simultaneously to adjust the marketing strategies accordingly with aim to improve customer experience. Understanding customer intent in real-time is considered vital to improve data-driven decisions. Machine learning offers the framework to build models that can identify the objective of a user. Many studies have utilized it and managed to predict purchase intention with high accuracy. None of them took the advantage of online machine learning that allows the models to be updated continuously without retraining needed. This study utilized probabilistic, linear and tree-based online machine learning methods with goal to achieve this, using a well-known experimental dataset from UCI ML repository. In addition, the impact of features dimensionality on the models' performance examined. The models evaluated in terms of AUC, sensitivity and specificity and the results suggested that online classifiers are considered promising to achieve this task, with tree-based model pointing out an overall better performance.

Eleftheria Trigeni

2023

Contents

List of Tables

List of Figures

1 Introduction

E-commerce is growing in a rapid manner and in the future is expected to develop even more. COVID-19 pandemic caused a significant impact on the e-commerce industry and that led to a significant growth in online shopping. The lockdowns and social distancing measures which implemented around the world accelerated the shift from traditional to online retail. In 2021 the worldwide e-commerce reached the amount of 5.2 trillion dollars, and it is predicted that in 2026 it will reach the value of 8.1 trillion dollars worldwide [1].

The pandemic also affected consumer behavior and people are spending more time researching products and comparing prices. This results in the phenomenon of individuals visiting an online retail platform but ultimately engaging in browsing and product research rather than completing a transaction [2]. High shipping costs, expensive prices and insufficient payment methods are some of the grounds that can lead to website abandonment. In addition, a common occurrence is to add products in their virtual cart but finally decide not to buy them. This situation is called cart abandonment [3]. Customers add items to their online shopping carts for various reasons other than purchasing intention. Nevertheless, saving products for feature purchase, or creating a list with the favor products are some factors that lead consumers to add items in the cart [3]. Furthermore, transaction abandonment is commented at [4] which describes the phenomenon where consumers have reached the payment stage but finally do not finish the transaction.

Thus, it is comprehensible that determining a customer's buying intent can be a complex task. Businesses are unable to determine whether the customer are likely to return to complete an order in the near feature or if they are going to make the purchase from a competitor, so this presents a significant challenge for any e-commerce business. Conversion rate in e-commerce, that is a key performance indicator which measures the ratio of visitors who turn into buyers [5], is estimated between 1 - 4% on average worldwide [6]. The low conversion rate is a hurdle for every business that aims to increase its

revenue. To address this issue, e-commerce website owners try to find ways to increase this ratio and turn more browsing into buying.

Machine learning methods offer the framework to predict and analyze customer behavior in real-time [7]. By making good use of these insights, businesses can better understand their customers, improve data-driven decisions, and adjust their marketing strategies accordingly. Machine learning also can continuously learn and adapt to changing customer behavior, making it a vital tool for predicting, responding to customer constantly changing needs and identifying causes that lead customers to leave the website without making a purchase. Acting towards all customers the same way is not effective. Consumers have become more demanding and want a personalized experience. If they do not "win" this experience in this e-commerce, then they will look for it in another [8]. Understanding the customer intention can be useful for promoting personalized recommendations, offers and improving customer experience.

In the language of machine learning, purchase intention prediction could be considered as a binary classification problem, where the goal is to predict whether a customer proceed to a purchase, considering several features regarding the customer and its behavior during the interaction with the website. Commonly, a classification algorithm be trained on historical data to learn patterns and relationships between the features and the target variable, and then be used to make predictions on new, unseen data [9].

Machine learning can be classified into two main categories, offline (batch) and online learning. Offline machine learning classification models be trained on a fixed amount of data and that means that all the data must be available in once. This approach does not allow the models to update during the training process and therefore they cannot catch changes in data over time. In contrast, online learning classification algorithms can learn and adapt themselves as new data arrive. They are designed to make predictions on new data on fly and simultaneously learning from these new observations [10].

Online learning is considered challenging, but it is useful since it adapts continuously in the new data and in e-commerce sector, being able to update the model incrementally is crucial to keep in track customer behavior. The real-time characteristic of this approach allows companies to quickly respond to customer needs and make data-driven decisions to convert a consideration of purchase to an actual transaction. Online learning is used mainly for streaming data, meaning data that generated continuously [11]. Large e-commerce platforms are visited constantly by users and all these users generate a huge

volume of data that should be process on time in to be beneficial. Indicatively, in May of 2022 Amazon, the most well-known e-commerce platform, announced 2.4 billion visitors [12]. This volume of data highlights the need for building algorithms that have the potential to handle such amount of data using as much as fewer resources.

In this study, the data generated from the users while they visit the e-commerce are considering as streaming data and online learning algorithms are applied on these in order to build models that can address customer needs in real time. This approach allows for continuously learning users' data and learning the recent patterns, identifying any changes in user behavior while browsing in order to retain accurate results over-time.

Literature showed that online machine learning classifiers seem to perform very well when have to face streaming data and produce comparable results with the offline learning techniques [10,14,23]. Various applications of online machine learning have been found in the literature including intrusion detection [15,16,17], human activity recognition [14], fake news detection [18], network traffic [19], improvement of wireless communication systems efficiency [20], handling sensor data produced by IoT devices [21,22] and process large continuously updated databases such as educational databases [23].

Although, as far as I am aware, none of the studies in the literature utilized the potential of online machine learning to predict purchase intention and this will be the contribution of this study. Linear, tree-based, and probabilistic classifiers are going to compared about their ability to distinct users with buying intent from those with browsing intent. Various feature sets are going to be fitted the classifiers with aim to find the optimal feature set and to examine the impact of the number of features on the learning process of each classifier. The findings of this study suggest that online machine learning provides promising results in the purchase prediction tasks. However, this study contains some limitations that will be discussed in the end providing a basis for feature research and improvements.

1.1 Problem Definition

E-commerce gains ground continuously, and businesses are looking to increase their revenue, so understanding user behavior in real-time is essential for taking actions to improve the customer experience. Identifying customers who are likely to abandon their purchase is crucial for customer retention, while recognizing those who are likely to proceed with a purchase can enable suggestions for additional purchases and boost revenue. Machine learning offers a valuable tool for identifying customer intentions in real-time. Existing approaches in the literature mostly use traditional machine learning methods, which are static and cannot adjust to changing customer needs over time. However, there are also dynamic methods that use Deep Learning to adjust accordingly and have achieved great results. There is a lack of implementation of online machine learning to predict purchase intention, which is promising due to its ability to be updated continuously and therefore it adapts to constantly changing customer behavior. Various online classifiers, including linear, tree-based and probabilistic models, exist, and this dissertation aims to implement these models to accomplish this task. In addition, these algorithm will be fitted with different feature sets to examine their performance with various number of features.

1.2 Research questions

To clarify the goal of this study, the research questions are listed below:

1) How online classifiers perform in the purchase intention prediction task?
2) Among tree-based, linear-based and probabilistic classifiers, which perform better?
3) How the number of features affect the online classifiers performance and learning procedure?

1.3 Dissertation Outline

The study is structured as follows. Chapter 1 provides an introduction and the problem statement. The fundamental part including definitions and background concept is presented in Chapter 2. Chapter 3 contains a summarization of similar works found in literature review. A description of the experimental dataset is presented in Chapter 4. Chapter 5 contains the methodology followed and the tools used for this study. The results of the implementation of online algorithms are presented in Chapter 6. The last part of the study, Chapter 7, discusses the conclusions, limitations and suggests feature improvements.

2 Backround

2.1 Machine Learning and its impact in E-commerce

Machine learning methods offer the framework to teach computers how to utilize data, learn from them and identify useful patterns. With the rapid increasing of available data, machine learning gains significant popularity. There are many categories of machine learning algorithms including supervised learning, unsupervised learning, semi-supervised learning, deep learning, ensemble learning and reinforcement learning etc. The focus of this study is on supervised learning that is a commonly used technique. Within the framework of supervised learning, labelled data are needed to fit the model meaning that the algorithm accepts input while the output is already known. Supervised algorithms handle these data and learn the relationships between input and output and then expand their knowledge to predict the output of new unseen input data. Prior training a supervised algorithm, the available data are divided into two sets: training and test set. Training set is exploited from the algorithms to learn the patterns between input and output and the test set is used to evaluate the ability of the classifier to generalize its knowledge in new data. The evaluation is mandatory before the model being deployed into production to ensure its efficiency [24].

Businesses take the advantage of machine learning techniques to understand their data and improve their performance making good use of them and taking data-driven decisions. Agriculture, healthcare, smart cities, and e-commerce are only some of the domains that make use of this tool [25].

In the field of E-commerce, businesses use machine learning to understand customer behavior while they interact with the website. Marketing is based on customer data with aim to offer a customized customer experience, retain the existing ones and create campaigns to attract new ones [26]. It is crucial for any business to retain its customers as well as to obtain new ones. Therefore, both kind of customers are profit for the business. On the one hand, retention of existing customers is vital for any business, as less cost is required to retain a current customer than to acquire a new one. Loyal customers

are more likely to make additional purchases in the near future. On the other hand, acquiring new customers contribute also to business success since it helps to expand the customer base and increase revenue [27]. The latter scenario is more challenging, but it is feasible utilizing technology and adjusting marketing strategies.

However, marketing strategies are costly, and it is vital to be targeted. Therefore, while a customer is interacting with the e-commerce, it is crucial to identify his intent meaning determining if he engages with the website with aim to complete a transaction or not. Customers who do not proceed to a purchase usually can be classified into two categories: either from the time they landed on the website did not consider spend money either they landed with a purchase desire but ultimately, they are not satisfied with the products, the prices, the shipping costs, offers, or cannot find what they search for [28].

That's when marketing actions need to intervene in real time to optimize the experience of these consumers by promoting personalized actions. For example, this could be done by chatbots that are used to improve the customer assistance, asking for what looking for and how could it help. Chatbot can be a kind of Artificial Intelligence so a machine treating like a human and immediately responds to customer needs trying to keep him on the website [8]. Personalized offers and offers with limited expired time may also be made to match consumer preferences.

Understanding customer needs in real time is crucial for the construction of recommendation systems which utilizing machine learning techniques, recommend products to the users based on their journey on the website so far, increasing the likelihood of purchase and is considered a productive technique [8,28]. They can be used to suggest products to the customers that just browsing as well to the customers that are going to complete a transaction by proposing items for additional purchases taking the opportunity of revenue increasing.

It is therefore noticeable that a company being able to understand the customer the moment they are active, contributes to the proper management of each customer individually and this in its turn can contribute to increasing the conversion rate.

2.2 Offline and Online Machine Learning

Machine learning can be categorized into two major categories based on how the models be trained: online learning and offline (or traditional or batch) learning.

2.2.1 Offline Machine Learning

In the context of offline machine learning all the required data for model training and evaluation are being collected before it comes to process them and therefore, during handling them, all are available. That means that during the training process of a machine learning model, the model considers all the available data and adjusts its parameters accordingly. Moreover, there is the possibility to access the data more than once. Then the model be used for making predictions on new unseen data [19]. It is generally believed that training a model in offline mode results into better performance.[29] A drawback of this approach is that when the distribution of the new data arriving changes, the model may no longer be so effective and to retain its accuracy, retraining it again in on all the available data is essential. That is time consuming, computational expensive and assumes that there is enough memory to store all the data [30]. With the rise of big data where data continues to expand and change rapidly, the limitations of these approach become more intense [31].

2.2.2 Online Machine Learning

Online learning refers to a more dynamic approach where the data come gradually and be processed one instance or one batch at a time. This allows models to update their parameters incrementally in every step as soon as new data arrive ensuring that they are always ready for predictions at any time. Regarding classification, similar to the offline learning, online learning can be used to accomplish supervised tasks, meaning that all the labels of the new instances are available, semi-supervised tasks where only a proportion of labels are available during the learning procedure and unsupervised task where no available labels exist [31].

In this study we are interested in supervised learning where labels consist of two classes indicating if a session ended up with a purchase or not. The algorithm of online supervised learning described in [10,33] as follows (Algorithm 1):

Algorithm 1: Supervised online learning

Algorithm 1: Supervised online learning

1: Initialize: $w_1 = 0$

2: for t=1,2,..,T do

3: The learner receives an incoming instance: $x_t \in X$;

4: The learner predicts the class label: $y_t = \text{sgn}(f(x_t, w_t))$;

5: The true class label is revealed from the environment: $y_t \in Y$;

6: The learner calculates the suffered loss: $l(w_t, (x_t, y_t))$;

7: if $l(w_t, (x_t, y_t)) > 0$ then

8: The learner updates the classification model: $w_{t+1} \leftarrow w_t + \Delta(w_t, (x_t, y_t))$;

To summarize, the classifier starts with initialize weights and for every data point data it accepts, it predicts the label, then it learns the true labels, it calculates the loss which is the distinction between the actual and the predicted value and it adjusts its parameters accordingly.

In case that data come in batches, the batch size is defined, let say n, and so for every n instances, the classifier updates its parameters. A drawback of this approach is that the batch size should be defined from the user. In addition, the classifier cannot learn from the latest samples before the batch size be fully collected and that may lag the classifier ability to adapt to the new data on time [34].

There is some confusion with the definition of online machine learning and incremental learning. In the literature, some studies identify these terms while other studies distinguish online learning from incremental learning by defining online learning when the data come continuously and process one instance at time and by referring incremental learning when data come in batches and be processed in batches in real time [19,35,36,37,38]. There are also studies that call as instance-based incremental learning when data come one by one and incremental batch-based when data come in batches [39,34]. Moreover «life-long learning» has been used as definition to describe online

learning [40]. To clarify, in the sequence of this study, online learning will refer to the phenomenon of processing one instance at a time.

Online learning solves the problem of changing on data distribution over time since the algorithms learn continuously. In addition, this technique addresses the problem of data storage since data are processed on fly and do not need to be stored. Online learning algorithms are well-suited for handling large-scale applications where data increase constantly and come with high speed [31].

In the following table (Table 1) some differences between online and offline learning are concentrated to highlight the differences of these learning methods:

Table 1 : Offline versus Online Machine Learning

Offline Learning	***Online Learning***
1. Data are available in once 2. Possibility to access the data more than once 3. The model needs retraining overtime including new data to retain high accuracy 4. Offline algorithms cannot adapt into changing of data distribution. 5. Require more resources since data should be stored	1. Data come continuously over time 2. Access the data only once 3. The model is updated as new data instances are coming and no retraining is needed 4. Online algorithms can adapt into changing of data distribution. 5. Data are proceeded on fly so limited resources are required

2.2.3 Data Streams

Data which come with high speed and continuously are defined as data streams or streaming data and need to be handled in real time [41,42]. Streaming data can be classified as stationary or non-stationary [43]. Stationary data streams are considered those who's the distribution does not change over time meaning that the statistical properties remain constant. On the other hand, in the context of non-stationary data, distribution can change over time. The change in data distribution is known as «concept drift» [42,43]. Concept drift is commonly handled by two approaches. The first approach involves updating the model continuously even there is not a concept drift every moment. The second approach is to use methods that identify if there is a concept drift or not and update the model accordingly. In addition, in data streaming scenarios, an approach

commonly used is to maintain a window with the more recent data and gradually as it slides to discard the older ones. So, in every iteration of the model implementation, it considers the statistics only from the data that are in the current window and is updated accordingly [42,43].

The process of mining those data in real time and extracting intelligence information that can help the decision-making practice is known as data stream mining. There are challenges come into the light when it comes to talk about data stream mining, including the inconsistency of those data since they come continuously, not uniformly and without a stable rate, making it difficult to analyze and find useful patterns. In addition, automated preprocessing techniques are needed to handle those data. Another issue is the large volume of data while the memory size is limited. So, there is a key trade-off to analyze large volume of data in limited memory and resources and simultaneously quickly in order to improve decision making in real or near real time [45].

Batch machine learning techniques cannot face these challenges and the importance of online machine learning techniques is highlighted to handle this scenario with efficiency. Online machine learning techniques gain ground in various data streams applications such as analyzing transactions, clickstream data, sensor data, stock prices, fraud detection, real time recommendations, network management, etc [42,46].

2.1Classification Models

Several algorithms that are used for classification purposes exist including linear, probabilistic, tree-based and ensemble methods. Literature shows that the performance of these algorithms depends on the specific task and requirements. That's highlights the need for experiment with different classifiers to achieve the optimal results.

For this study, the online learning classifiers Hoeffiding Tree, Hoeffding Adaptive Tree and Extremely Fast Decision Tree selected as tree-based models. Moreover, Logistic Regression with Stochastic Gradient Descent optimization and Passive Aggressive choosed as linear classifiers. Finally, Naive bayes that is a probabilistic classifier also included.

2.1.1 Hoeffding tree

Decision trees have gained popularity in many traditional machine learning applications [48]. They use a tree structure to classify the instances based on the input features. [49]

It is consisted of nodes, leaves, and branches [50]. Root node is located on the top of the structure [51]. Every node depicts a feature, every brunch illustrates a rule, and every leaf demonstrates an output. In the context of classification, the output are the possible values of the target variable. During the training process the algorithm, based on the input features of each instance, evaluates the decision rules associated with each node taking into consideration some splitting criteria and thresholds and this way it follows a path in the tree with aim to reach a final output value that is the final class of the target variable. It is considered easy interpretable by humans due to its simple rule-based and decision-making approach [50]. During the training of a decision tree, the algorithm uses all the available data in order to determine the optimal path that leads to the final class of each instance. When it comes to handle streaming data, due to the fact that the data come gradually and are not available in once, this methodology is not suitably. Hoeffding Trees introduced by Domingos P. and Hulten G. as an alternative incremental approach to face this issue [52]. Hoeffding Trees (HT) could learn from data streams and assume that the distribution of the incoming data remains constant as time passes [52]. Introducers based their idea on the Hoeffding bound which mathematically calculates the minimum number of instances that is required for the accurate statistics estimation. Considering this, they support that a relatively small sample size can be sufficient to select the best appropriate feature (node) for splitting [53].

2.1.2 Hoeffding Adaptive Tree

As abovementioned, Hoeffding Trees assume that the data distribution does not change over time. However, as abovementioned, in the context of streaming data this is not always guaranteed. Hoeffding Adaptive Tree (HAT), is an alternative approach of Hoeffding Trees that uses Adaptive Window Method (ADWIN). ADWIN is an algorithm which developed to detect concept drift and is considered as a widely recognized technique to address that issue. This method uses an adaptive sliding window, meaning that its size changes dynamically. Specifically, it raises its size when there is not a concept drift. Differently, it decreases its size [42]. The decisions which are taken to classify each instance are based on the current sliding window which contains the most recent instances. The implementation of the algorithm has as idea that when a concept drift is detected, a candidate subtree is grown [55]. This way, the algorithm adapts on any changes on the data distribution.

2.1.3 Extremely Fast Decision Tree Classifier

Extremely Fast Decision Tree Classifier (EFDT) introduced by Manapragada, Webb and Salehi [56] and it is another altered version of Hoeffding Tree. Both algorithms use similar approach but differ in the way they split the node. Hoeffding Tree algorithm finds the optimal split and this decision does not change. In contrast, Extremely Fast Decision Tree Classifier, when it finds a candidate split at a node, it splits it, but in the sequence, it reconsiders the decision and if a improvable split occurs, then it proceed at it. Extremely Fast Decision Tree Classifier overcomes Hoeffding Tree in terms of statistical effectiveness but lags in terms of compositionality [42,56].

2.1.4 Naive Bayes

Naive Bayes is a **probabilistic classifier,** and it is based on Bayes theorem [57].

Let say that $\bar{x}$ are the features of the dataset in a D-dimensional feature space and y is the target variable. In a binary classification task, y ∈ [0,1].

Then according to the Bayes Theorem [58]:

$$p(y = i|\bar{x}) = \frac{p(i)p(\bar{x}\,|\,i)}{p(\bar{x})}$$

where p(i) = prior probability of the class label

and p(y|$\bar{x}$) = probability distribution of both the features x and the labels

Naive Bayes simplifies the process assuming that the features are independent when conditioned on the class labels [58].

$$p(y = i|x) = \frac{p(i) * [p(x_1|i)p(x_2|i) \dots p(x_D|i)]}{p(x)}$$

The assumption of the independence consists a limitation, however despite that, Naive have achieved excellent outcomes in various machine learning applications. [59,60,61,62,63,64]

Naive Bayes is well known for its straightforwardness, requires minimal computational effort and is considered suitable for incremental tasks [65]. In the process of online learning, Naive has the ability to learn quickly when only few data are available and that results in accurate predictions from the early stages [66].

2.1.5 Passive Aggressive

Passive Aggressive (PA) classifier was first introduced by Koby Crammer for binary classification purpose regarding online learning and hypothesizes linear separability [67]. In the beginning the weights are initialized, it receives an instance and makes a prediction. Then, it learns the true label of that instance. The classifier can update its weights accordingly by addressing three different versions of the optimization problem [68].

In case that the prediction was correct, it updates its weights in a passive way otherwise the weights adjustment be aggressive and hence the name Passive Aggressive [69]. The parameter C, namely «aggressive parameter», controls how much aggressive will be every update. Larger values of C result to more aggressive updates while small values lead to smoother adjustments [67]. The pseudocode of Passive Aggressive classifier described in [67] as follows (Algorithm 2):

Algorithm 2: Passive Aggressive Classifier

INPUT: aggressiveness parameter C > 0

INITIALIZE: w_1 = (0,...,0)

For t = 1,2,...

- receive instance: $x_t \in R^n$
- predict: y_t = sign($w_t \cdot x_t$)
- receive correct label: $y_t \in \{-1,+1\}$
- suffer loss: $\ell t = \max\{0 , 1-y_t (w_t \cdot x_t)\}$
- update:

1. set: $\tau_t = \frac{l_t}{||x_t||^2}$ (PA)

$\tau_t = \min\left\{C , \frac{l_t}{||x_t||^2}\right\}$ $(PA-I)$

$\tau_t = \frac{l_t}{||x_t||^2+\frac{1}{2C}}$ $(PA-II)$

2. update: $w_{t+1} = w_t + \tau_t y_t x_t$

2.1.6 SGD Classifier

Stochastic Gradient Descent (SGD) is an optimization algorithm for learning by minimizing a loss function [42]. In this study, SGD used to minimize the Logistic Loss. In the online learning, in each iteration, SGD updates the model's parameters in contrast with the offline learning that the gradient descent takes into consideration the whole training set in order to update the parameters.

In the first step of SGD, the parameters are initialized randomly. Then, the gradient of the objective function is calculated, and the parameters are adjusted based on the gradient. The magnitude of the update is controlled by a hyperparameter called learning rate. Large learning rate values lead to sharper updates, and this has the risk of exceeding the minimum of the cost function. On the other hand, small learning rate results to more iterations until convergence and as a result the training time is increased [42]. It is an optimal method for large-scale data where achieves great results [70,71,72].

2.2 Evaluation metrics

Evaluation of the classifier efficiency before it goes into production is a very important step of machine learning. Various classification metrics exist that measure its performance, including accuracy, true and false positive rate, precision, recall, f1-score, sensitivity, specificity and Area Under the Curve (AUC) [73]. Choosing the right metric is important to properly evaluate an algorithm as each classification problem has specific requirements.

Accuracy which is very well-known metric and measures the classifier overall ability to predict correctly counting the number of true positives and true positives predictions. The disadvantage of this metric is that it does not consider the class imbalance. Even we balance the dataset, this metric cannot provide a comprehensive representation of the classifier power to distinct the two classes [74]. One metric from the above mentioned useful for the binary classification task is the AUC. An interesting study which presented in [75] proved and explained why AUC metric is considering better than accuracy. Also, according to the findings of the same study, Naive Bayes algorithm and tree-based algorithms often have same accuracy score but differ in AUC score and so it is vital to take it into account in order to evaluate these algorithms precisely.

AUC or Area Under the Roc Curve is a representative metric for ROC curve which is a plot that shows the relationship between the true positive rate and false positive rate for

different classification thresholds [76]. AUC values range from 0.5 to 1.0, where 0.5 indicates that the classifier make predictions as a random classifier while 1.0 suggest a perfect classifier. With simple words, AUC is a metric that reveal the ability of a classifier to separate the two classes of a binary classification problem. For the task of purchase intention, AUC measures the potential of a classifier to distinguish sessions that are going to end up offering a revenue to the business and those who are going to end up with only browsing.

In addition, in this work, classifiers evaluated in terms of specificity and sensitivity. Sensitivity or True Positive Rate (TPR) or recall refers to the ratio of correctly positive predictions (TP) divided by the total number of positive samples while specificity, or True Negative Rate (TNR) refers to the ratio of correctly negative predictions (TN) divided by the total number of negative samples [77].

The selection for these evaluation metrics was driven considering that in e-commerce sector it is crucial for every business identifying in real time both users who are likely to abandon and those who have intent to complete the transaction. For this study, sensitivity is the proportion of purchases that correctly identified by the classifier and similar specificity is the proportion of sessions that are going to end up without revenue that correctly classified. In our task it is crucial to achieve high specificity because detecting these events, allows businesses to provide targeted offers of limited time, personalized recommendations when the users are still active, or promoting email marketing aiming converting them into buyers.

On the other side, determining sessions that are going to end up with a purchase is also important as this presents an opportunity to businesses to increase their revenue. For example, due marketing strategies, similar products for addition purchases can be suggested in real time.

2.2.1 Evaluation of online classifiers

Evaluation of online classifiers is regarded as a challenging task and various approaches have been proposed in the literature to accomplish this. Although the evaluation metrics are almost the same with the offline classifiers' ones', the method that they are applied differ.

A commonly used approach are the prequential metrics (either prequential AUC either prequential Accuracy). Many studies including [78,79,80] used prequential metrics to

examine the efficiency of their models. Prequential AUC firstly investigated by Brzezinski, D., & Stefanowski, J in 2015 [79]. The idea for AUC generated because it is able to evaluate efficiently even the dataset is imbalanced. Prequential AUC measures the model efficiency based on the most recent data, using a sliding window. As new data added to the window, the old data are gradually discarded from the window and forgotten. This method ensures that the evaluation is always representative for the most recent data.The algorithm 3 (retrieved by [79]) describes in detail the pseudocode which is used to calculate the prequential AUC in a stream.

Algorithm 3: Prequential AUC

Algorithm 3 Prequential AUC

Input: S: stream of examples, d: window size
Output: $\hat{\theta}$: prequential AUC after each example 1:
W ← ∅; n ← 0; p ← 0; idx ← 0;
for all scored examples x t ∈ S do
    // Remove oldest score from the window
    if idx ≥ d then
        scoreTree.remove(W[idx mod d]);
        if isPositive(W[idx mod d]) then
            p ← p − 1;
        else
            n ← n − 1;
        // Add new score to the window
        scoreTree.add(x t);
        if isPositive(x t) then
            p ← p + 1;
        else
            n ← n + 1;
        W[idx mod d] ← x t ;
        idx ← idx + 1;
        // Calculate AUC
        AUC ← 0; c ← 0;
        for all consecutive scored examples s ∈ scoreTree do
            if isPositive(s) then
                c ← c + 1;
            else
                AUC ← AUC + c;
        $\hat{\theta}$ ← AUC/ pn ;

Other studies [78,80] use also ADWIN prequential evaluation metrics meaning that instead of defining a windows size, they use ADWIN which has the ability to adjust the windows size dynamically. Furthermore, there are studies which keep a separate test set of the stream and calculate AUC as in the offline mode including [81,82] while calculating the cumulative AUC considering the whole stream also has used [83]. A comparison among these evaluation ways conducted in [84] and found out that prequential AUC is considered useful for identifying drift detection [79].

In this study, a combination of evaluating the classifiers in both online and offline fusion applied. A similar methodology also used in [30] and is described in section 5.

3 Related Work

3.1 Purchase intention prediction

Several studies found in the literature that focus on purchase intention. The researchers took the benefits of machine learning and deep learning to achieve the goal. A review of the studies is presented.

An interesting approach implemented by Esmeli, R., Bader-El-Den, M., & Abdullahi, H with aim to predict early purchase intention [85]. The authors of the study analyzed e-commerce log data to predict whether a user session would end up with a purchase. They extracted temporal and various other features based on user behavior and timestamps and used them to fit five machine learning models: Decision Tree, Random Forest, Bagging, K-Nearest Neighbors, and Naive Bayes. The dataset was quite imbalanced and that they addressed this problem with various techniques. The models evaluated in terms of AUC and a custom scoring function used to further assess the accuracy. The results suggested that among the five models, Decision Tree model was the most accurate. The duration of the session proved to be the most significant predictor of a purchase. A limitation is that the algorithm is limited to sessions where users have browsed two or more products and so is not able to make predictions if a user has only browsed a single product in the session.

An interesting work is presented in [86]. The authors took the benefit of deep learning to predict purchase intention in real time taking into consideration only user session-

based data. Their approach was dynamic means that for each user session the algorithm predicts, in each user step, if the remaining session would end up with a purchase or not. Their results based on RNN-LSTM model which achieved a very high accuracy. They also tried to predict which sessions that had items in the cart did not convert to transactions but due to the limitations of such sessions the performance of the model was poorer. The strong point of this study is that their approach is dynamic and simultaneously it does not need any feature engineering and that allows updating the model, in real world applications, without limitations in case it is needed. In reference [87] a study that has similar nature is presented. The authors aimed to predict in real time the outcome of a user session by utilizing the clickstream data generated during the session. By training a RNN-LSTM, the authors classified the outcome of a session as either browsing only, cart abandonment, or purchase. To optimize the performance of the model, they experimented with various sequence lengths in the RNN training process. The use of clickstream data and the implementation of an RNN model concluded to impressive results.

Separation between browsing and buying session is crucial for any e-commerce. In the paper [88] researchers experimented using real data from an online bookstore. After the construction of 23 attributes related to user sessions, they trained SVM models with different kernels with aim to predict whether the session would end-up with a purchase or not. Even though their dataset was quite imbalance, the SVM classifier with linear kernel managed to predict the buying session with probability 95%. A similar approach implemented by Ahsain, S., & Kbir, M. A in [89] using a dataset with 18 attributes. Especially, they trained various models like Light BGM, Gradient Boosting, Random Forest and the tuned versions of them. Random Forest with tuned parameters had the best performance and achieved an accuracy of 91%. Both studies managed to predict purchase intention with a very high accuracy. However, none of them is dynamic and this is a limitation.

In a study conducted by the authors in reference [90], several classification models were compared to predict sessions that would result in a purchase. After models' evaluation, it was determined that the Random Forest algorithm achieved the highest accuracy of 89.55%. It was also found that the implementation of Gradient Boosting led to a further increase in accuracy to 90.34%. Based on these results, the authors concluded that Gradient Boosting is a promising approach for predicting sessions that would end up with a purchase. Researchers of reference [91] also agreed that Gradient Boosting algorithm is

the ideal for this task since it was the winner among 7 classifiers with an accuracy of 91%. Despite the fact that both studies accomplished valuable outcomes, the lack of adaptable behavior consists of a limitation since the model cannot adapt in customer needs over time.

Another approach for predicting purchase intention implemented in the paper in reference [92]. Utilizing session data, the authors constructed a purchase intention model and a system aiming to predict likelihood abandonment. For purchase intention prediction Multilayer Perceptron, SVM and Random Forest were chosen to overcome the task. Also feature selection took place based on correlation, mutual information and mRMR filters and top ranked features fitted incrementally the training models. After experiments they found out that class imbalanced was crucial to achieve better results. Multilayer Perceptron with ten hidden layers achieved to predict purchase intention with an accuracy of 87.24%. The top 6 features that suggested from mRMR contributed to this achievement.

In reference [93] an interested study is presented. The most important part of this work is that the researchers proposed an ML system capable to predict purchase intention upon visiting the e-commerce, before interacting with the products. This approach is crucial for any e-commerce since it allows for immediate action to retain a customer. They experimented in 3 datasets which all had low conversion rate and studied registered and unregistered users separately. Random Forest was found to be the best classifier in terms of accuracy; however Decision Tree predicted the most TP samples. The performance of the models was better for registered users who had historical data that determine their behavior and previous purchase found to be the most important feature. Despite the fact that their study was very detailed and achieved great results, it does not take into consideration the change in users' behavior over time.

In paper [94] deep learning and machine learning approaches compared according to their ability to predict if a session would end up with purchase or not. Deep learning models found that are more accurate for their prediction problem. According to their work, and among session-based and customer-based data, the time that the session started, the day of the week, the session duration, and the number of days between last purchase are the most useful attributes for the task.

Also studies that aim to predict purchase intention when add products to cart found in the literature. Rausch, T. M., Derra, N. D., & Wolf, L [95] used clickstream data from a

German retailer which consisted of significant attributes about users' behavior. After applying descriptive analysis, they discovered that, on average, those who proceeded to a purchase viewed and added more products to their shopping carts compared to non-purchasers. The study conducted a detailed comparison between various classification models and the researchers found that Gradient Boosting with regularization outperformed the other models, resulting in an accuracy of approximately 82% in predicting cart abandonment. Another similar recent approach is presented in [96]. A machine learning system created with aim to restrict the cart desertion. Evaluating various classifiers, CatBoost model had the best performance (accuracy 76%) and it is based mainly to the total visits of the user to the specific e-commerce, the duration that the customer visits the site, the total purchased sum and also the maximum and the minimum product price that the customer has viewed during his visit. They also used the method LIME to interpret the model prediction and justify how it led to a decision and this is a strong point for the study since interpretability is vital for any e-commerce.

Various studies have examined customer purchase intention in e-commerce, using a variety of techniques and valuable insights have been extracted. Many studies have shown that models in this task have better results when they use only the most informative features. However, to the best of my knowledge, none of the studies takes the benefits of online machine learning that allows the continuously learning and predictions in real time. Only studies that utilize sequential models like RNN-LSTM found in the literature that are dynamic. Adapting and predicting in any time is crucial for e-commerce since customer needs and behavior change over time and the models should adapt quickly in these changes and provide immediately results. So, it seems to be a gap in the literature, and this is an opportunity for further examination of the purchase intention.

3.2 Online Machine Learning Techniques

In addition, a variety of studies also found in the literature that utilize online classification algorithms to accomplish different classification tasks. They specialize especially in their performance in training time and the ability of a classifier to make right predictions. Studies that compare the performance of online and offline classifiers have also been conducted. A brief of these studies follows.

Various online classification algorithms compared based on their ability to predict network traffic in the study [19] of reference. Researchers in their work, included 5 ensem-

ble-based algorithms (Adaptive Random Forest, Very Fast Decision Rules, Online Boosting, Additive Expert and Oza Bagging) and in addition VFDT, HAT and Extremely Fast Decision Tree. They experimented in 3 datasets and the evaluation based on Accuracy and Kappa metric. Between the ensemble classifiers, all had a precious performance, but Adaptive Random Forest was the winner in terms of accuracy (99%). They noticed that all the classifiers, in the beginning of the process presented an instable behavior but as more sample were coming, the performance was getting steadier. Among tree-based algorithms Extremely Fast Decision Tree achieved the highest scores while Very Fast Decision Tree performance was the worst. They found out that ensemble techniques can have higher ability to predict network traffic. Another work [97] found in the literature that discuss extensively the classification of data streams. Hoeffding Tree, Naive Bayes, KNN, Adaptive Random Forest, Leverage Bagging and Active Learning compared in terms of accuracy, evaluation time, precision and recall. They experimented in 3 datasets and according to their findings, Hoeffding Tree achieved the highest evaluation metrics. One more comparison conducted in [98]. The researchers compared multiple online classifiers for their ability to predict fake news. Passive Aggressive classifier and SVM outperformed in terms of accuracy while Random Forest, Logistic Regression, Naives Bayes and SGD had poorer performance.

The researchers of the reference [99] introduced AB-HT which is an ensemble incremental algorithm for intrusion detection system. AB-HT is a combination of AdaBoost and Hoeffding Tree. It has the ability to preserve the accuracy of a model without any retraining needed which makes it less computational expensive. After experiments they found out that it takes less time to train the proposed model rather than AdaBoost-Decision Tree. In addition, the model achieved significantly higher F1-score than HT and HAT models, making it a propitious choice for intrusion detection.

Authors of [46] conducted a comparison between batch learning and streaming learning algorithms in their study. As online classifiers they utilized Hoeffding Tree (HT) and OzaBagAdwin (OBA) while for the batch mode they used Decision Tree (J48), Projective Adaptive Resonance Theory (PART) which is an algorithm uses a partial decision tree technique to imply rules. Experiments were conducted for both binary and multiclass classification tasks. Among the online classifiers, in both tasks, OBA outperformed in terms of accuracy but required more training time. According offline learners, J48 provided the best results. The researchers, comparing both offline and online meth-

ods, led to the conclusion that online classifiers outperform traditional ones' in the binary task while the latter had better performance on the multiclass classification task.

A comparison between batch and online machine learning classification models conducted by the authors of the paper in reference [101]. Their aim was to predict the user response time in the Stack Overflow and they considered it as a binary classification problem. They compared five batch models including Decision Tree (DT), Logistic Regression(LR), Support Vector Machine (SVM), k Nearest Neighbors (k-NN) and Deep Belief Networks (DBN) with three online models including Stochastic Gradient Descent (SDG), Perceptron, and Passive- Aggressive (PA) in terms of accuracy and training time performance. They found out that DBN was the winner in terms of accuracy (66.5%) among all the models. However, all online learners outperformed in terms of training and prediction time as they were able to produce results in as little as 3000 times less time compared to the batch models and that highlights the dynamic of online learners when data come continuously and with a large volume. Among the online classification learners, the highest accuracy achieved by SGD (63.6%) while PA and Perceptron performance was poorer. Ling Kock Sheng and Teh Ying Wah [102] seems to concur that training a classification model in batch mode results to better accuracy. This was the finding in their study focusing on comparison between batch and incremental classification techniques while predicting credit cart risk. Several models fitted with bank data and evaluated in terms of accuracy. Specifically in their case Decision tree overperformed a Naïve bayes updateable model. The evaluation of models restricted to accuracy and did not consider the training time and the prediction of the model. Researchers of reference [103] conducted a study aiming to predict human activity recognition. One of the objectives of their work was to compare online and traditional classification models. They found out that online models achieved comparable accuracy to those of batch learning models. Although, the required training time was significant shorter.

Authors of the paper [104] studied the difference between batch-incremental and instance-incremental. They experimented in multiple real and synthetic datasets and also with multiple windows sizes. They evaluated batch classifiers like: SVM, DT, LR, and online classifiers like Naive-Bayes, Hoeffding Tree ensembles, SGD and k-NN in terms of accuracy, time and ram-hours. After various experiments they concluded that instance implementations have a similar performance with the corresponding batch im-

plementations but require fewer resources and that is a significant factor in streaming data. In reference [23] another comparison between instance and batch incremental classification conducted. The authors experimented with educational data which require incremental learning since new students enroll continuously into the educational system. Five models were fitted both in batches as well as new data arrive. Also they applied ensemble learning with aim to increase the accuracy of the models. Although the training time was not examined, in terms of accuracy all the models reveal that can achieve accurate results either in batches either in instances.

Online Naive Bayes classifier with exponential weights for moving average and standard deviation studied in [105]. The implementation relied on fact that the last samples that a classifier see should weight more in order to keep updated and retain its performance over time. After experimentations for a binary classification problem on a well-known dataset the researchers found out that Naive Bayes outperformed Perceptron in terms of accuracy but lacks on behind KNN classifier. However Naive Bayes requires less resources than KNN and that's an advantage.

The study conducted in [106] suggests interesting results. Authors examined the influence of ensemble methods for Network Intrusion Detection. They studied both homogeneous and heterogenous methods where homogeneous refer to combining several equal models while heterogeneous meaning combining varied models. For this work, KNN, SVM and Hoeffding Adaptive Trees (HAT) utilized for the construct of heterogeneous ensemble model. Adaptive Random Forest (ARF) combining multiple Hoeffding Trees (HT), Boosting of HT and Boosting of HAT included in the homogeneous group. The evaluation based on prequential AUC. Firstly, the algorithms examined for their ability to predict correctly each one separately before proceeding to the ensemble methods. HT outperformed in terms of AUC but SVM was the quickest algorithm. KNN was much more consuming from the other algorithms. Among homogeneous classifiers, ARF with 20 HF trees was the winner since it resulted to an AUC of around 100% on their experiments despite the fact that required double time of that of ARF with 10 trees. Among heterogeneous, the combination between HAT and ARF was the best. They concluded that ensemble of both categories can enhance the performance of the models till up to 8% but require more time so it is a trade-off between these two factors.

Researchers of reference [107] aimed to examine the efficiency of HAT in the intrusion detection using a well-known dataset. In order to test its performance, they compared it with other classifiers including KNN, Naive Bayes and Perceptron. They evaluated the algorithms after every 1% samples of the dataset and in the end they calculated the average for each metric. Perceptron had a very poor perfomance while Naive Bayes had slightly better results. Although, KNN and HAT seemed to be the most promising one for this task since they reached average and precision more than 98%. KNN was the most time consuming, followed by HAT.

A range of online algorithms including tree-based, linear-based, probabilistic, ensemble and lazy classifiers, found to accomplish different tasks. in the cases compared with offline, they had comparable results. In general, in each task a different algorithm achieves the best results.

4 Experimental

4.1 Data Source

The experimental dataset «Online Shoppers Intention» was taken from Machine Learning Repository [108], a repository with various datasets for research purposes. The dataset contains information about customers browsing behavior. It is consisted of 12,330 sessions of unique users, 15,5% of them ended up with a purchase. Various studies used this dataset in order to predict purchase intention [92,109,110,111,112,113] using various machine learning and deep learning models. The number of attributes equals to 18 and «Revenue» is a binary target variable indicating if a customer made a purchase.

For the research purpose, we assume that the data come in a streaming fusion, without taking into account the chronological order, so the models learn gradually and simultaneously make predictions.

The below table (Table 2) concentrates the features of the dataset:

Table 2: Dataset description

Feature	*Description*
Administrative	The number of pages visited by the user in the administrative section of the website
Administrative_Duration	Amount of time the user spent on administrative pages
Informative	The number of pages visited by the user in the in the information section.
Information_Duration	Amount of time the user spent on informative pages
Product Related	Number of product related pages that the user visited
Product Related_Duration	Amount of time the user spent on product related pages
Bounce Rates	The percentage of visitors who enter the website through that page and then left the website without engaging in any further actions
Exit Rates	The percentage of pageviews on the website that end at that specific page
PageValues	Average value for a web page that a user visited before completing an e-commerce transaction
Special Day	Whether it was a special day
Month	Month
Operating System	An integer value that indicates the type of operating system used by a user while accessing the webpage
Browser	An integer value that indicates the browser used by a user while accessing the webpage
Region	An integer values represents the region of the user
TrafficType	An integer value indicates what type of traffic the user is classified to
Visitor Type	An integer value represents if the user is new
Weekend	An integer value representing if the session occured on weekend
Revenue	An integer value representing if the session ended up with a purchase

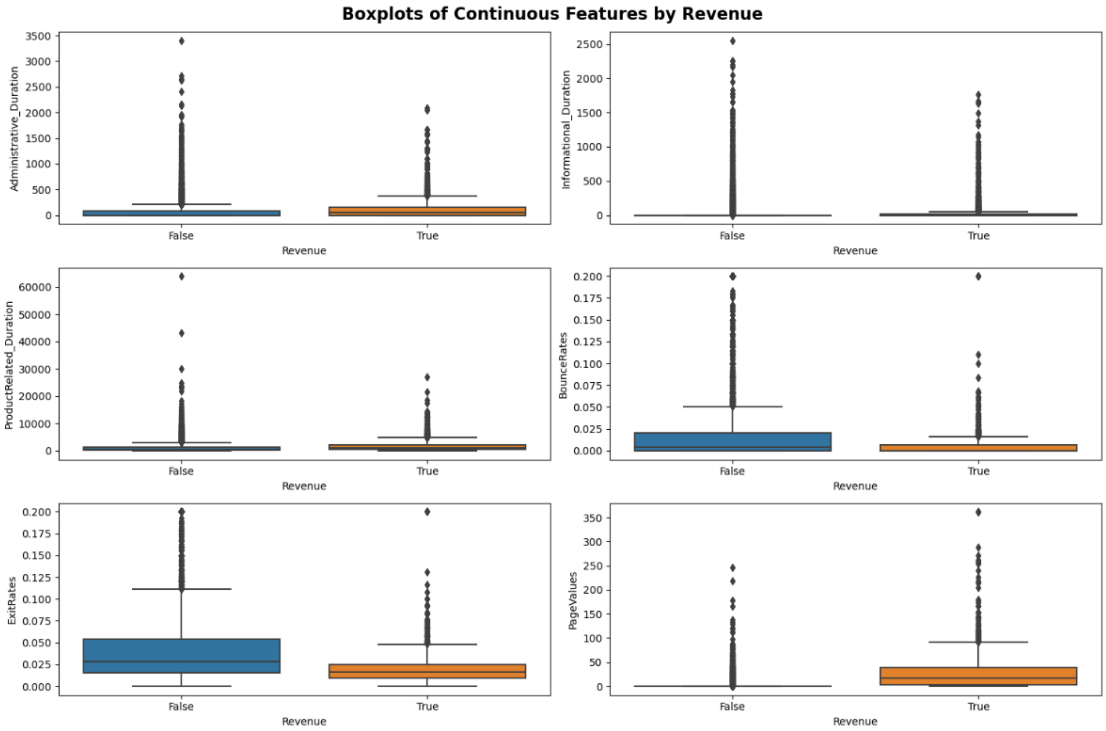
«Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" are attributes which obtained from the URLs of the pages visited by the users during their sessions. «Exit Rate», «Bounce Rate» and «Page Value» are Google Analytics metrics. Google Analytics track website traffic by collecting data and generate report with valuable insights that marketers can utilize to capture the performance of their websites [114]. «Bounce Rate» is a crucial metric for every e-commerce platform since it indicates the interest of a page that users landed on. High values of Bounce Rate may suggest that users are not satisfied with the content of a particular page and so they abandon it. In contrast, lower values of Bounce Rate point out that users interact with this page [115]. «Exit Rate», with simple words, shows how many times a particular page was the last one during a session and so high values of exit rate for a particular page indicate the poor contribution of a particular page [116]. Finally, «Page Value», measures the participation of a page to the business revenue. In this dataset, «Bounce Rate», «Exit Rate» and «Page Values» refer to the average of these metrics of all the pages visited by the user. By making good use of all these metrics, conversion rate and revenue can be increased.

4.2 Descriptive Statistics

Some descriptives statistics are provided to take a more comprehensive understanding of the dataset.

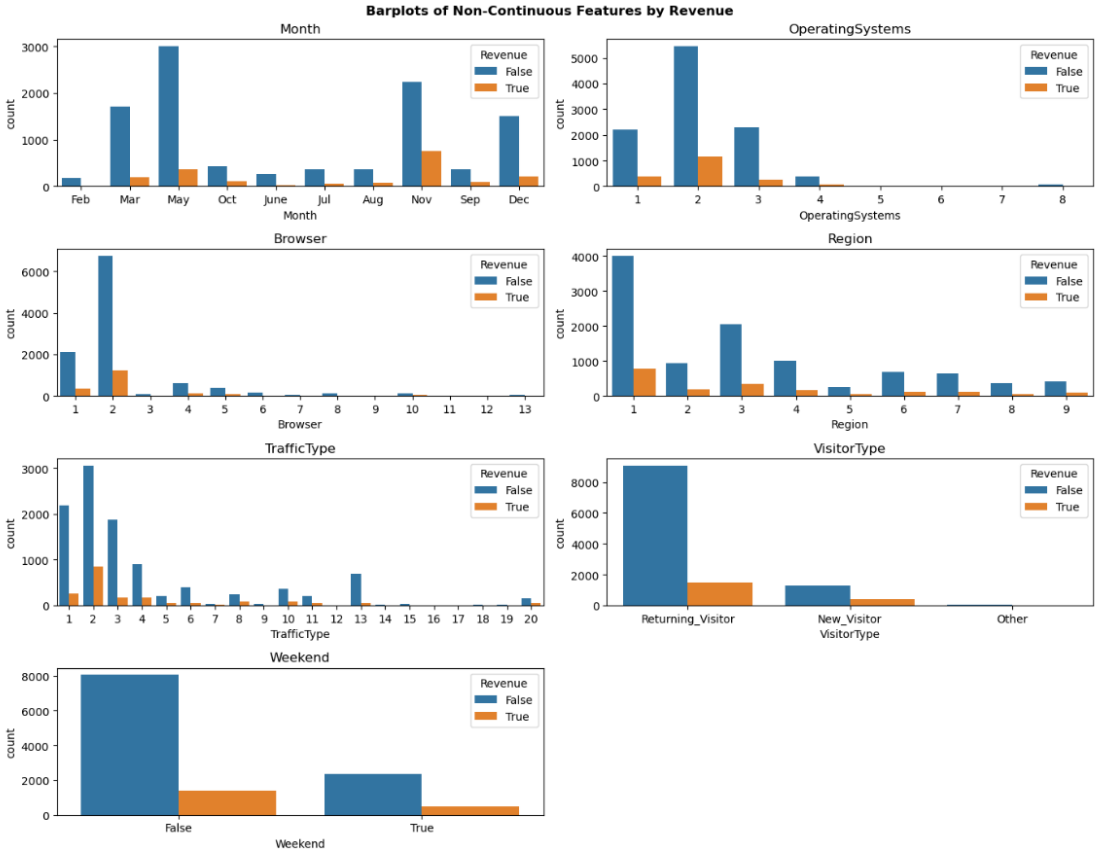
The below figure (Figure 1) illustrates the relationship between the target variable and the continuous features of the dataset. It suggests that the attribute «PageValues» is going to be a good predictor for purchase intention task. Most sessions that end up with a purchase seem to have much higher page values. It does not make sense since, «PageValues» measures the contribution of a page to the revenue generated by the website. The remain Google Analytics metrics, «Bounce Rate» and «Exit Rate» seem to support the prediction too.

Figure 1: Boxplot of continuous features by revenue



The association between Revenue and Non-Continious features is displayed in the below figure. (Figure 2). By examing it, there are some insights that we could take advantage of. According the month, most purchases took place on November. Operating system «2» seems to be the most promising one for the business revenue and the same applies for browser «2» and Traffic Type «2». In addition, region 1 gives more turnover to the business. Finally, most purchases have been done by returing visitors and that's highlight the importance of retaining customers in a business.

Figure 2: Boxplot of non-continuous features by revenue



To get a better understanding of the variables it is crucial to examine the correlations between them. The correlation matrix (Figure 3) suggests strong correlations. A high positive correlation exists between «ExitRate» and «BounceRate» which indicates that when one of these metrics inscreases, the other increases too. In addition the strong correlation between «ProductRelated» and «ProductRelated_Duration» is a sing that the most «Product Related» pages a customer view, the most time spending of them and vice versa. Figure 4 and Figure 5 confirm the linear relationship between these pairs of variables. The same applies and to the other type of pages but the relationship is not so strong.

Figure 3: Correlation matrix

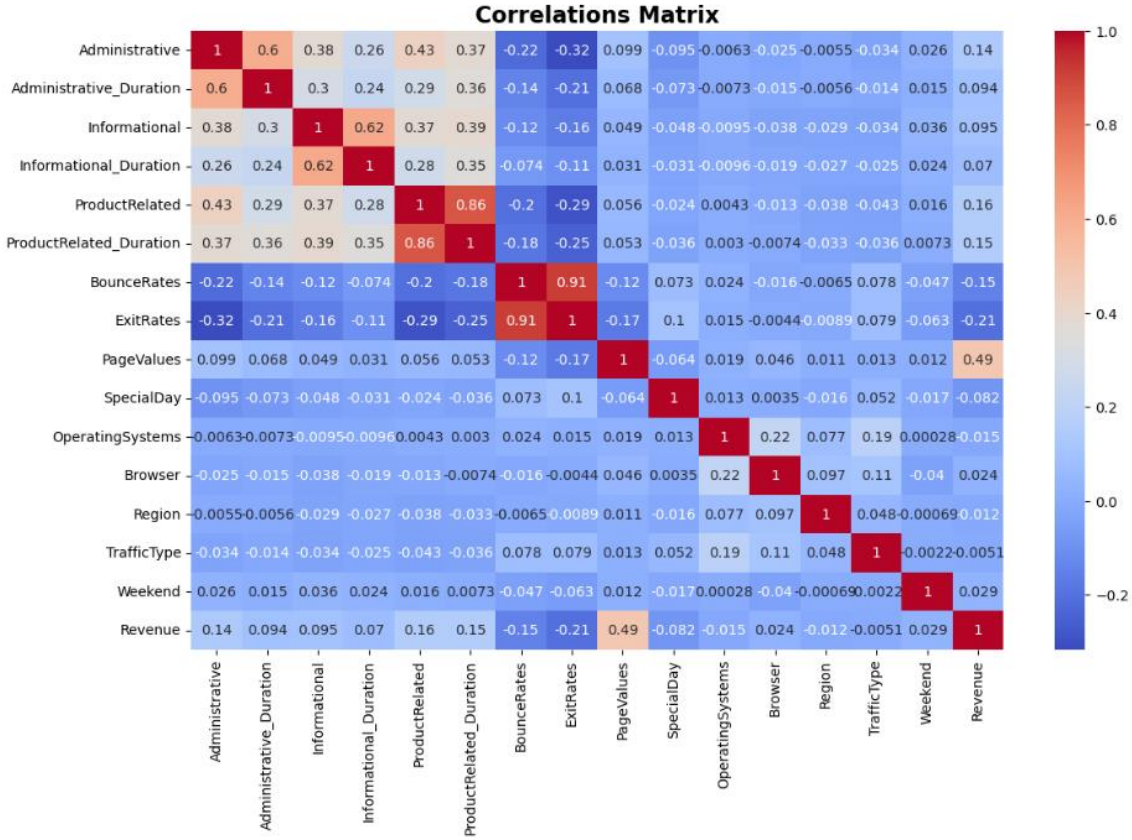


Figure 4: Relationship between ProductRelated and ProductRelated_Duration

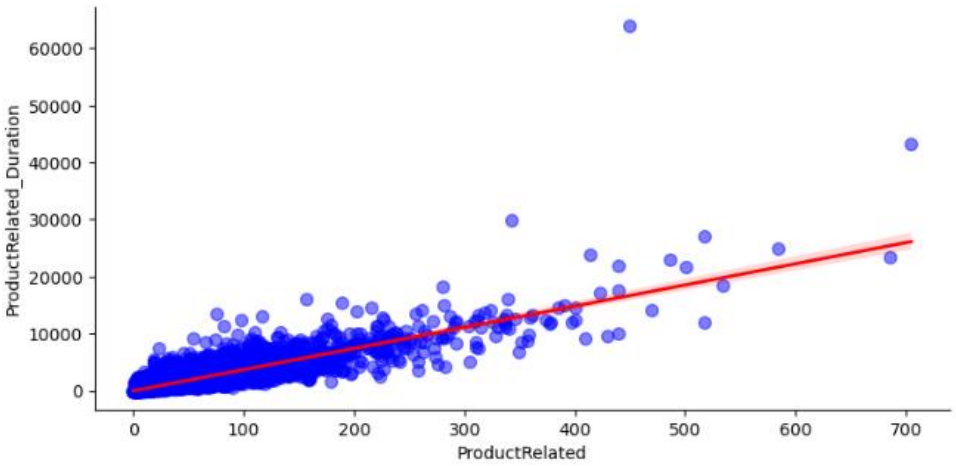
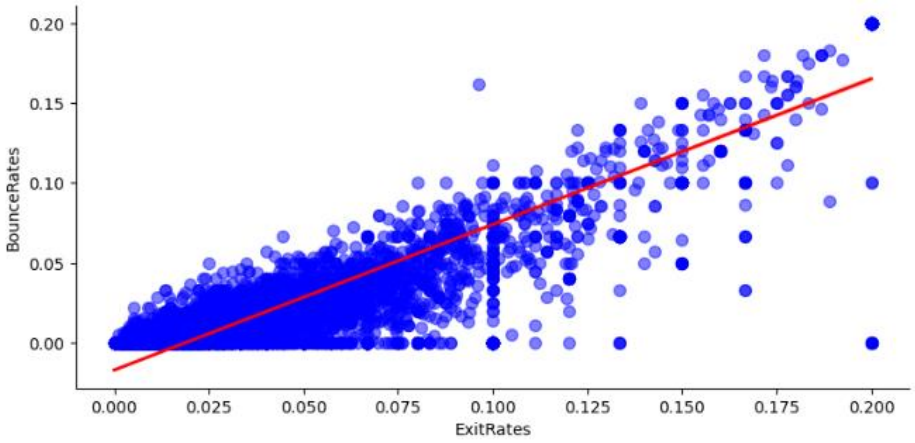


Figure 5: Relationship between ExitRates and BounceRates



4.3 Data Preprocessing

The dataset does not require a lot of preprocessing in order the features to be able for the classification task in the sequence. However, certain steps were essential. In particular, the categorical variables "Month," "VisitorType" "Weekend," and "Revenue" were encoded into numeric values. Furthermore, the dataset splitted into training set and test set. Training set consists of the 80% of the whole dataset and is used for model training and real time predictions while test set consists of the 20% of the whole dataset and is used to evaluate model performance on unseen data. The ratio of purchases events maintained in both sets in order the datasets to be comparable and have a better sense for the model's performance. In other words, stratified sampling applied. In real streaming data scenarios, it is not feasible to have a test set, however in this study we assume that the test set is a ratio of available data that is used to examine the generalization ability of the classifiers.

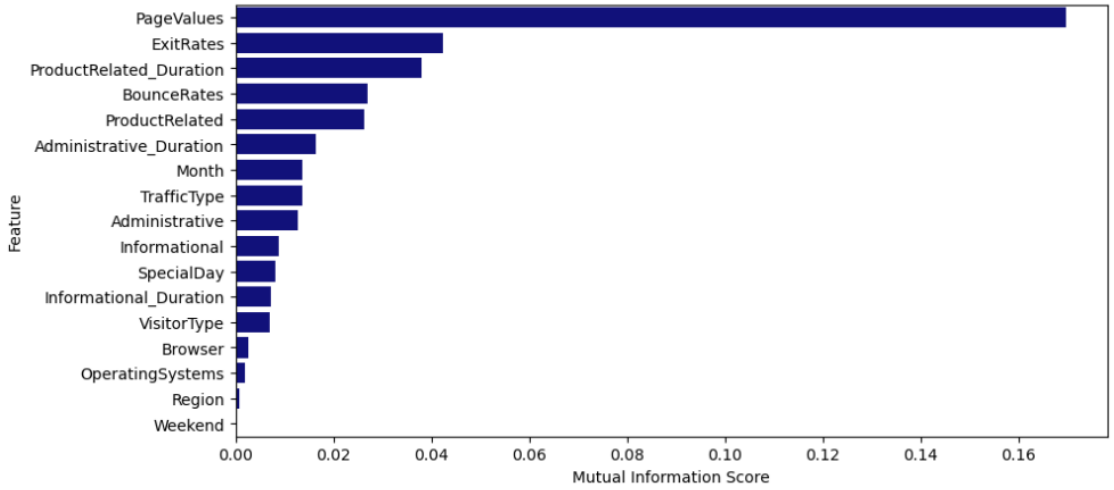
4.4 Feature Importance

To achieve high performance in a classification task, retaining the most informative features is crucial as suggested by studies in the literature. By doing this, model's performance gets better and simultaneously training time is reducing. Various feature selection methods appear and have been used in the literature. Sklearn package in Python provides several algorithms to accomplish this. In this study «SelectKbest» method selected to assigns scores regarding the mutual information to each feature. The below figure (Figure 6) illustrates the results obtained. It is confirmed that «PageValues»

seems to be the most important predictor since its importance assigned to this feature is much higher than the others. «Weekend», «Region», «OperatingSystems» and «Browser» are considered the less informative variables in terms of mutual information.

Feature selection done in offline mode since online feature is considering more challenging. This is a limitation in this study and it is suggested for further work. However, in the sequence different feature sets according to their importance are utilized in order to examine how the online classifiers power is affected by the number of features.

Figure 6: Features scoring based on mutual information



5 Methodology

This section describes the methodology used to train and evaluate the machine learning models aiming to recognize the outcome of a session.

5.1 Tools used

For the implementation, the programming language Python used. Libraries like matplotlib, seaborn and pandas and plotly used for manipulating, preprocessing, and visualizing purposes. The library «River» that is well known for online machine learning contributed at most. «River» library built based on two popular libraries Creme and scikit-

multiflow that support also online learning. River package provides multiple possibilities for both batch online and instance online learning. However, instance online is its strong point since it provides a variety of methods that support this approach. It is used for popular tasks like classification, clustering, regression, and anomaly detection. [117]

In terms of classification that is applied in this study, river provides various classifiers including linear, tree-based, naive-bayes, neighbors and ensemble methods. Each classifier has the corresponding hyperparameters that can be tuned to improve the performance of any task.

5.2 Deal with imbalance

Several techniques are available to overcome the hurdle of class imbalance. For this task random oversampling of the minority class was chosen to deal the problem and balance the two classes to have equal weights. In batch learning where all the data are available, random oversampling means adding samples in the minority class to reach the desired class distribution. In the context of online learning where data come continuously, a different approach is applied. River package provides the «RandomOverSampler» method that is designed to work with streaming data. In particular, «RandomOverSampler» is a wrapper that during each iteration, considers the distribution of the two classes that the classifier has seen so far and adjusts the samples accordingly [118]. Another attempt to address the issue conducted with under-sampling the majority class or combining the two methods. However, the results of these two approaches were similar and had not an improvement so oversampling selected to address the issue of class imbalance. An attempt also conducted, that is presented in the next session, to train the models without handle imbalance in to examine how each of them perform in this situation.

5.3 Standarization

Standardizing the data to have zero mean and unit variance is a common technique that is applied in order to improves the model's performance. When all the data are available in once, the mean and the variance is known so subtracting the mean and deviding by the standard variation give the standardized data. However, in online machine learning the process slightly differs. «River» package provides the method «StandardScaler()». Instead of mean, a running mean is calculated as new data arrive. Similar, instead a

standard deviation, a running standard deviation is calculated incrementally. So in simple words, mean and standard deviation are updated in each iteration and the data are adjusted accordingly [119].

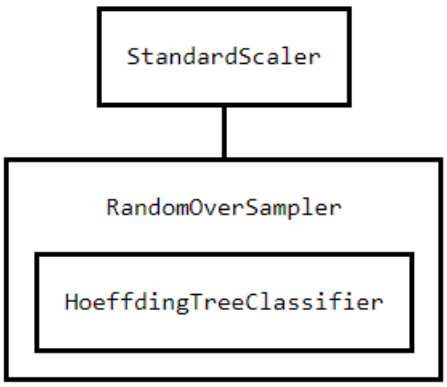
5.4 Pipeline

River also provides Pipeline method that simplifies the code by concentrating multiple steps in one command. In this work, a pipeline used to combine standarization, random oversampling with equal desired distribution and the classifier (Figure 7 and 8).

Figure 7: Constructing a pipeline in river package

```
model=(
    preprocessing.StandardScaler()|
    imblearn.RandomOverSampler(
        classifier=tree.HoeffdingTreeClassifier(),
        desired_dist={0: .5, 1: .5},
        seed=42
    )
)
```

Figure 8: Visualizing a pipeline including Hoeffding Tree Classifierr



5.5 Hyperparameter Tuning

Hyperparameter tuning is essential in order to achieve better results in any machine learning task. For linear models that are simpler, testing the performance of models for different parameters is a simple way to select the best model. However, when dealing

with streaming data this is not always possible since data arrive continuously and can proceed once [120] but this method could be applied in the initial iterations to check models' efficiency. Here for the study purpose and because of limited data, we compare hyperparameters assuming all the data are available. This approach applied to Passive Aggressive Classifier for the aggressive parameter (C) and to SGD Classifier for the learning rate (lr) and the results are presented in the following chapter, chapter 6.

5.6 Train and make predictions

Independently the classifier, the current library follows a simple way to train and evaluate the models. The method *stream.iter_pandas* is used to simulate the pandas data frame as a streaming setting. Every row of the dataset consisting of the features and the target variable. The method *stream_iter_pandas* is called in every iteration to read one row of the dataset at a time. The model predicts the label of the features coming using the method *model.predict_one*. Then, it uses the method *model.learn_one* in order to learn from this instance and so it adjusts its parameters. This process is applied in a for loop until the whole dataset has been processed. This manner, the model makes predictions in real time and learns sequentially. The algorithm is described in (Algorithm 4):

Algorithm 4: A simple implementation of learning and predicting process in river

Algorithm 4: A simple implementation of learning and predicting process in river

```
1. auc_metric = metrics.ROCAUC()
2. for i, (xi, yi) in enumerate(stream.iter_pandas(X_train, train_label, shuffle=True, seed=1)):
3.         y_pred = model.predict_one(xi)
4.         auc_metric.update(y_true=yi, y_pred=y_pred)
5.         model.learn_one(xi, yi)
```

5.7 Evaluate the models

Moniroting the model performance in real time

Prior the for-loop (Algorithm 4) AUC evaluation metric is initialized and after each iteration it is updated. So, after each iteration, the cumulative AUC is known and be visu-

alized. By doing this and monitor the performance of the model in real-time we have an overall comprehension of the classifier's efficiency so far. The purpose is to examine the learning procedure of the models and their progress in their ability to distinct the two classes over time. A question also generated is how the models of different groups (tree-based, linear-based, probabilistic) are affected by the number of features they fitted with.

Evaluate the model offline on the test set

During the online learning procedure, sequential models are constructed. Each model, predicts the label of the next instance, and then being fitted with the actual label. Then a new model is constructed based on the information of this instance. This process is repeated until all instances pass. After n iterations, n models have been constructed and each model is an update version of the pervious one. The nth model is the one that have learned all the information from all the previous instances. Evaluating this model on the test set allows for examining its generalization ability [30]. Here, specificity, sensitivity and AUC value as considered since we are interested to see the models' ability to generalize and identify both promising and browsing customers.

Taking into account both real-time and offline evaluation metrics, gives an universally idea of the model power. Noteworthy that in the online setting, because the AUC metric is cumulative and takes into account even the first observations where the classifier has not yet learned, we will only get a first impression of the classifiers' performance and then they will be evaluated in the test set where we will get more detailed conclusions.

As authors of [30] claimed, in case that two classifiers have the same performance on the test set, the one who achieves higher cumulative metrics during the online procedure, means that capture earlier the patterns of the data, or in the language of machine learning, it converges earlier.

Each business can select the best model based on the evaluation metric that meets its needs. Note here that each business has different needs, and it is important to target with high probability the customers of interest because marketing campaigns are costly.

A business may want to focus on identifying as more as possible customers which have purchase intention. So, in this case, sensitivity is the metric of interest. Targeting them with proper real-time marketing movements like recommending products for additional

purchases is a good practice to take the best of these customers. On the flip side, identifying browsing customers allows the business to keep them engaging with the products, improving the customer experience and provide special personalized offers with aim to convert them into buyers.

Other businesses may would be interested in the AUC value of the model meaning identifying with high probability kind of customers.

6 Results

This chapter presents the main part of this study, that aims to predict purchase intention in real time using online machine learning approaches. To accomplish this objective, various techniques implemented with goal to improve models' performance and the results are presented.

Two linear models that abovementioned (Stochastic Gradient Descent and Passive Aggressive) three tree-based models (Hoeffding Tree Classifier (HT), Hoeffding Adaptive Tree Classifier (HAT), Extremely Fast Decision Tree Classifier (EFDT)) and one probabilistic model (Naive Bayes) compared in their ability to distinct sessions that are going to end up with a purchase with those that are not. All the models trained incrementally, meaning that they were updated as the new data were arriving and simultaneously predictions were made in real time.

Note that Naive Bayes is an incremental algorithm that assumes independency among the features. In the literature, it achieved promising results in many tasks. In the experimental dataset there are strong correlations. However, Naive included to the research with aim to examine its performance even the correlations, how they affect it, and which is the impact of features dimensionality on this classifier.

6.1 Training and evaluation on the original dataset

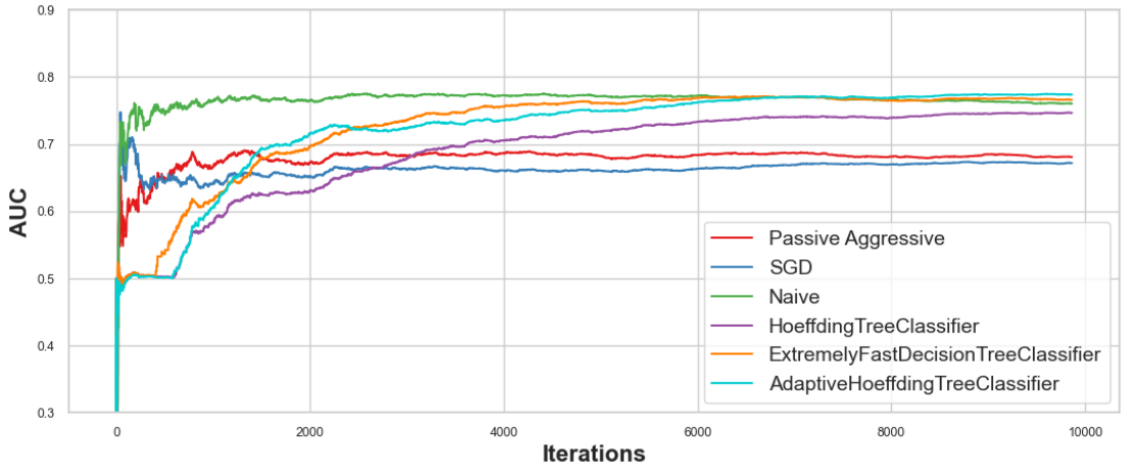
Even though the dataset is imbalanced (only around 15% of the visitors proceeded in a purchase), it is significant to check the online models' performance in the original dataset before applying hyper tuning and techniques that deal imbalance.

For that implementation, all the models contain the default parameters of the river package (aggressive parameter of PA equals to 1, learning rate for SGD equals to 0.01)

During this process, the AUC metric was tracking.

The below figure (Figure 9) illustrates all models' performance in the original dataset over iterations in terms of AUC during the real-time predictions and learning phase.

Figure 9: AUC over iterations in the original dataset



The above figure (Figure 7) indicates that the linear models behave differently than the tree-based ones in the predicting-learning phase, but the linear ones show similar behavior to each other, so do the tree-based ones to each other. Naive Bayes that is a probabilistic model behave differently from the two other groups of models in the original dataset. In the early iterations, the models have access on limited amount of data and their performance improves gradually as new data arrive and learn from them.

Naive Bayes seems to learn faster since after about 1000 instances he has seen, his AUC score starts to stabilize meaning that in 1000 iterations it has learned what it could learn and its performance neither improves nor decreases as new data are coming. This finding confirms [66] that via experiments found out the ability of this classifier to predict with accuracy from the early iterations.

Respectively, the linear models (PA, SGD) capture the patterns of the dataset after about 1500 iterations and consolidate their performance, but their results seem to be unsatisfactory. Also, in the beginning of the process, a lot of fluctuations are presented in contrast with tree-based models. The latter present a more progressive improvement in their performance and learn continuously for about 5000 iterations and perform quite effectively if we consider the class imbalance.

The below table (Table 3) illustrates the models' efficiency on the test set.

Table 3: Evaluation on the test set prior oversampling

		Evaluation on Test Set		
		AUC	**Specificity**	**Sensitivity**
Linear Models	**SGD**	65.57%	97.89%	33.25%
	PA	66.19%	89.97%	42.41%
Probabilistic Model	**Naive**	73.77%	82.10%	65.54%
Tree – Based Models	**HT**	74.60%	94.48%	54.71%
	HAT	71.89%	95.87%	47.91%
	EFTD	72.71%	95.15%	50.26%

All models have a relative low performance in terms of AUC. Linear and Tree-Based models are unable to determine the positive class (successful sessions) that is the minority class of the dataset, while their predictions are very accurate regarding the negative class, but this is a result of overfitting. Tree-based models seems to perform slightly better than the linear-based models on the imbalanced dataset in terms of sensitivity. The latter can predict the purchase events quite worse than a random classifier. The highest AUC values is observed by HT (74.60%) but its sensitivity is not satisfying.

None of these two groups of models could be used in a real scenario with such a poor performance. However, it is very important to point out that only the Naive Bayes Classifier managed to distinct the two classes to some extent despite the imbalanced of the dataset and that is an indication that Naive Bayes is not as sensitive to overfitting compared to other classifiers in this scenario and this happens despite the correlations that exist in the dataset. Nevertheless, the results prove that it is essential to balance the dataset to achieve better results.

6.2 Hyperparameter Tuning

This subsection presents the results of hyperparameter tuning. Noting that hyperparameter tuning applied including the oversampling wrapper in pipeline for more accurate results.

6.2.1 Passive Aggressive Optimization

The below figure (Figure 10) presents the AUC score over iterations for multiple values of the passive aggressive parameter C.

The plot indicates how the passive aggressive parameter affects the learning process of the classifier. For large values of C (C= 1 or C=0.1), the efficiency of the model during the real-time predictions gets much poorer, while the classifier gets more efficient when C=0.01 or C=0.001. Particularly, for C=1 or C=0.1 the classifier learns until around 1000 iterations and then there are observed some fluctuations which indicate that the classifier is not able to correctly predict the following instances and has not learn the patterns of the data. After around 2000 iterations it seems that the classifier has learned the patterns on the data but with poor performance especially when C=1.

On the other hand, when C=0.01 or C=0.001, the learning process of the classifier is smoother specifically in case that C=0.001. After around 3000 instances, the classifier has gained the information of the data, its performance stabilizes and achieves a relative high AUC value, denoting that it has achieved to distinguish customers with purchase intention from those with browsing intention in a sufficient degree.

This contrast among passive aggressive values happens because the updates of the model weights get more aggressive while C value increases, leading to faster updates but that often results to decreased effectiveness of the model on new data. On the other hand, smaller values of C, meaning less aggressive updates, delay the learning process but that results to more constant and accurate model. To ensure these indications and examine the generalization of these models, the model evaluated on the test set for the different values of C. Table 4 displays the results and confirms that the optimal value for the aggressive parameter of PA is 0.001. Remarkable the fact that the AUC value for C=0.0001 overcomes the corresponding of C=1 by 13% and which is a significant improvement and highlights the need to select the optimal passive aggressive value.

Figure 10: AUC over iterations for different values of passive aggressive parameter C

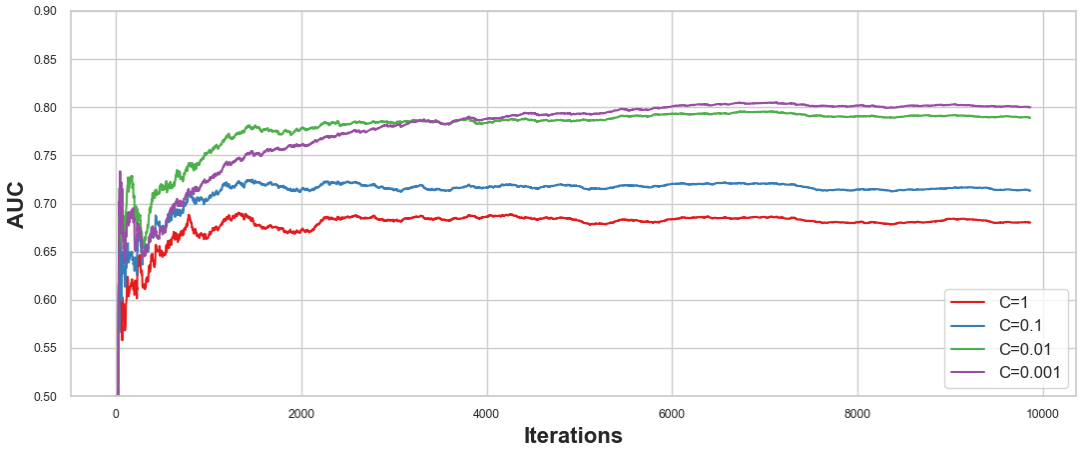


Table 4: Evaluation on the test set for various values of passive aggressive parameter C

Aggressive parameter	*AUC*
C=1	66%
C=0.1	70%
C=0.01	78%
C=0.001	**79%**

6.2.2 SGD Learning Rate Optimization

The AUC score over iterations for multiple values of the learning parameter of SGD classifier is illustrated at (Figure 11) while the table (Table 5) determines the efficiency of the algorithms on the test set. For large values of learning rate (lr=0.1 or lr=0.001), SGD classifier stabilizes its behavior earlier at around 2000 iterations. On the flip side, smaller values of learning rate (lr=0.001 or lr=0.0001) lag the training process and so the AUC value increases gradually. For lr=0.0001 the classifier learns throughout the whole training process.

It does not make sense since learning rate controls the step size in each iteration when it comes to update the model parameters and like the aggressive parameter of PA, small values of learning rate led to slight updates of the model weights after each iteration. For learning rate=0.0001, the AUC value after the classifier has seen all the instances, is the lowest one. That interprets by the fact that in this case, the performance of the model was improving in a small rate and so in the early iterations there were a lot of false pre-

dictions. Although, SGD with this parameter generalizes very well (AUC=76% on the test set) while during the real-time predictions the overall AUC value was 72%. On the flip side, when lr=0.1, the overall AUC during the training process was 74% but the model does not generalize quite well in the test (AUC=69% on the test set). That's confirms that the early converge of a classifier does not always guarantee good generalization. Learning rate=0.001 seems to be the optimal for this task since achieved the highest AUC value (79%) for both real-time predictions and test set.

Figure 11: AUC over iterations for different values of learning rate of SGD classifier

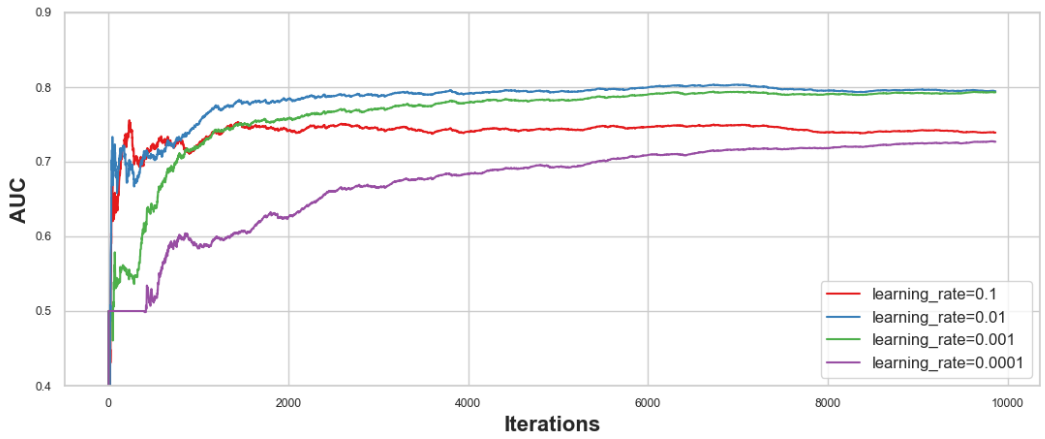


Table 5: Evaluation on the test set for various values of the learning rate of SGD

Learning rate	*AUC*
lr=0.1	69 %
lr=0.01	77%
lr=0.001	**79 %**
lr=0.0001	76%

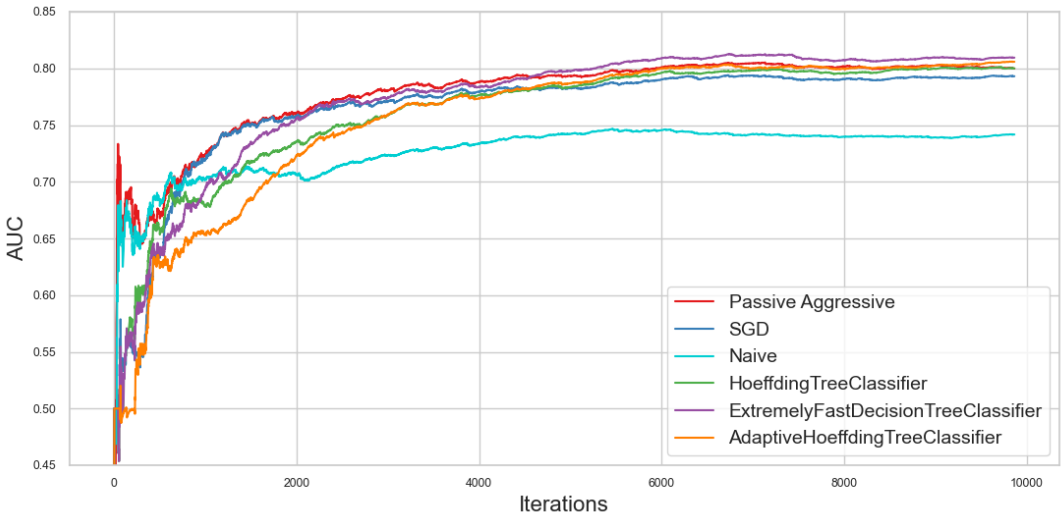
For the next experiments the classifier SGD has a learning rate lr=0.001 and the Passive Aggressive has an aggressive parameter C=0.001.

6.3 Train and evaluate after oversampling

After dealing with class imbalance and hyperparameter tuning, the same approach with the previous one applied for all the models, meaning training using online fusion and

simultaneously making predictions on the new data on fly. The effectiveness of the models was monitored during this technique in terms of AUC score and is displayed in Figure 12.

Figure 12: AUC score over iterations after oversampling



The efficiency of the models for real time predictions is much better for linear and tree-based models. In contrast, it seems that the overall performance of Naive Bayes did not improved and oversampling did not boost its ability for real time predictions. The remain classifiers achieve relative equal overall AUC after processing all the data (around 80%). The diagram shows in a degree the difference between linear and tree-based classifiers, since tree-based ones gradually increase their AUC, which means that they are constantly learning from new instances. However, because the diagram is confusing, the behavior of each classifier in this process will be discussed in detail below.

The below table (Table 6) concentrates the evaluation metrics in the test set allowing for a more comprehensive comparison of the models' generalization ability.

Table 6 : Evaluation on the test set after oversampling and hypertuning

		AUC	*Specificity*	*Sensitivity*	*Training and prediction time*
Linear Mores	**SGD**	79.37%	85.17%	73.56%	*4.34s*
	PA	79.43%	88.96%	69.90%	*2.94s*
Probabilistic Model	**Naive Bayes**	72.33%	63.77%	80.89%	7.56s
Tree – Based Models	**HT**	80.99%	89.20%	72.77%	7.63s
	HAT	83.18%	81.81%	*85.65%*	13.19s
	EFTD	80.61%	85.56%	75.65%	57.65s

After overcoming the setback of imbalance, even though the models differ in terms of architecture and computational complexity, they achieved comparable results on the test set. Both linear models resulted in similar evaluation metrics but PA lags in terms of sensitivity. In addition, both require a small amount of time and that is very beneficial.

Among tree-based models, also they have similar performance each other with HAT to overcome in terms of AUC value and sensitivity while HT in terms of specificity. EFTD achieved also promising results but required the most time to be trained.

Comparing all models to each other, tree-based models are the most computational expensive and especially EFTD. Indeed, EFTD requires around 18 times more time than PA that is the less time-consuming. It is worth noting that in this case the data set is small so all the times are relatively small. In real conditions where data is constantly increasing, the training time of a model is a very important factor so for EFDT this would be a disadvantage.

However, tree-based models have overall better performance in terms of AUC compared to probabilistic and linear models but even HT which is the fastest of all in the tree category, takes about twice time as long as linear models. To select the optimal model in this case, it is a trade-off between model effectiveness and training time, and it depends on the needs of a company but also on the resources it has.

HAT seems to have the highest ability to distinguish the purchase from the non-purchase ones' meaning that it has the highest AUC value (83.18%) followed by HT and EFDT. Considering this metric, Naive Bayes lags the remain classifiers. Indeed, when it trained without the oversampling wrapper, it achieves a similar AUC score

(73.77%) but with specificity= 82.10% and sensitivity=65.54% while now occurs almost the opposite (high sensitivity, low specificity). It seems that oversampling benefited its ability to identify the minority class but simultaneously significantly decreased its ability to identify the other class so after oversampling it is much stronger to identify customers that are going to offer revenue to the current business. This finding also highlights the need to consider several evaluation metrics.

HT overcomes among all the models in terms of specificity (89.20%) meaning predicting accurately the sessions that did not contribute to the business revenue. PA also achieved very promising specificity (88.96%). In general, all the classifiers seem to get well predicting these sessions, except of Naive that presents insufficient specificity (63.77%).

Regarding sensitivity, indicating the ability of the classifiers to identify customers with purchase intention, the highest value produces by HAT (85.65%) followed by Naive (80.89%). HAT also is more accurate predicting customers that are going to offer revenue than the browsing users. The other classifiers suggested not promising results regarding this metric.

6.4 Different features sets

Mutual information selector ranked the features according to their importance. PagesValue, ExitRates, ProductRelated Duration, Bounce Rates, ProductRelated, AdministrativeDuration, Month, Traffic Type, and Administrative are the top 9 features with «PagesValue» considering the most informative one. The classifiers selected to accomplish this task trained using various subsets of features to identify the best subset among these that contributes to the highest evaluation metrics. Also, another question generated is how the number of features affects the learning process of each classifier.

The following table (Table 7) displays the various subsets that were fitted into the classifiers.

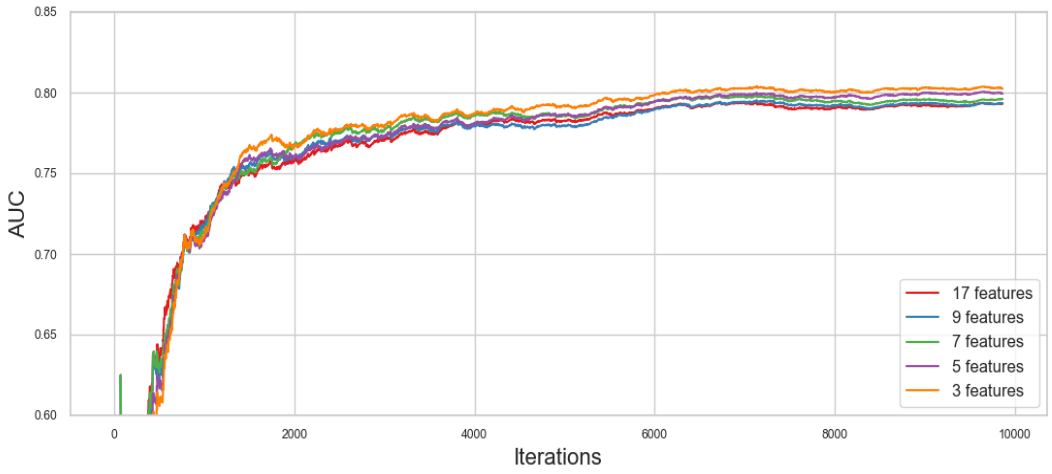
Table 7: Different feature sets that fitted into the models

	Features
Subset 1 (9 features)	PagesValue, ExitRates, ProductRelated Duration, Bounce Rates, ProductRelated, AdministrativeDuration, Month, Traffic Type, and Administrative
Subset 2 (7 features)	PagesValue, ExitRates, ProductRelated Duration, Bounce Rates, ProductRelated, AdministrativeDuration, Month
Subset 3 (5 features)	PagesValue, ExitRates, ProductRelated Duration, Bounce Rates, ProductRelated
Subset 4 (3 features)	PagesValue, ExitRates, ProductRelated Duration

The following graphs show how the models behave in terms of AUC in real time predictions with different subsets of features. The purpose of these plots is to see how each classifier learns in terms of AUC over iterations. It gives also an image of how many iterations it takes to capture the patterns of the data, and whether this depends on the number of features. In additional, fluctuations can be a sign of change in data distribution that a classifier cannot handle properly.

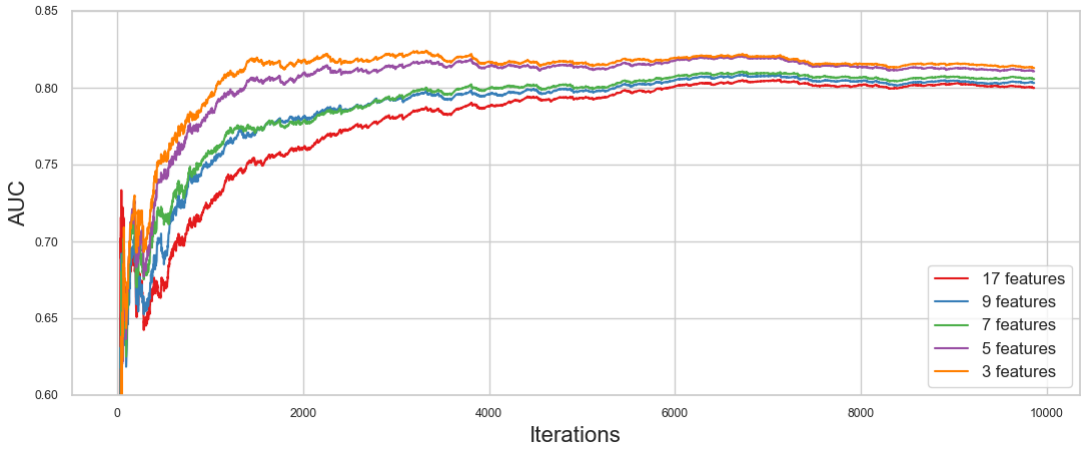
6.4.1 Impact of number of features in classifiers performance in learning and real time prediction process

Figure 13: AUC over iterations of SGD classifier with different number of features



SGD seems not to be affected in a highy degree from the dimensionality since the curve is almost same for all the combinations (Figure 13). Indepently the number of features the classifier learns gradually as new data come. Around for 2000 iterations, the AUC value increases continiously with a high rate indicating that SGD take the most information from the data. Then, it starts to increase in a lower rate meaning that it does not receive very informative data that can improve its performance even more and mainly this trend intensifies after 4000 iterations where its AUC almost stabilizes. After that, there are some fluctuations, points where the AUC decreases for a while and then increases again. For example this occurs around at 5000 iterations. Also shortly before 8000 iterations the AUC slightly decreases. These may indicate concept drift as new instances arrive. Regardless the dimensionality, SGD overall AUC after all the instaces has passed is about 80% meaning that it achieved to distinguish the two classes of the binary problem in a sufficient degree during the real time predictions.

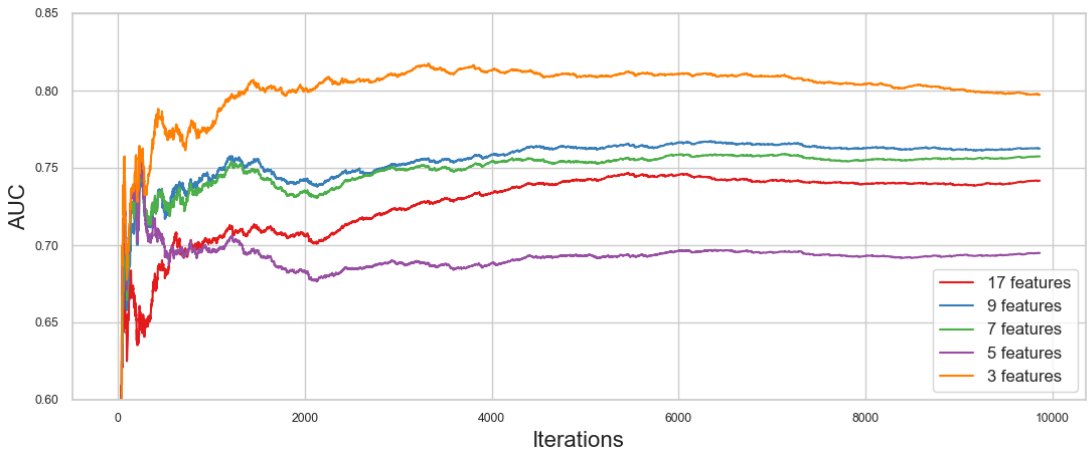
Figure 14: AUC over iterations of PA classifier with different number of features



PA (Figure 14) needs around 500 iterations in order to gradually enhance its effectiveness. It is reasonable that up to this point the classifier predicts almost ranodmly. From 500 until around 2000 iterations, it gains the ability to distinguish the two classes. We also notice a differentiation in terms of the number of features. With more features (17, 9 and 7) we notice that after 500 iterations its ability to separate the 2 classes increases, more steeply up to 2000 iterations and then more slowly up to 4000 iterations. After 4000 its AUC remains almost stable. In contrast, when it fitted in only 3 or 5 features, around in 2000 iterations it consolidates the AUC value meaning that

with less data the learning process is quicker and the ability to distinguish the two classes increases in the early iterations. However, the AUC values is almost the same after all the instances have been proceeded with only a slight improvement observed when fitting only three or five features. Similar with SGD, there are some fluctuations which indicate the the classifier cannot predict correctly these instances and could indicate a change in the data distribution.

Figure 15: AUC over iterations of Naive Bayes classifier with different number of features



Naive Bayes (Figure 15) present a completely different behavior. It seems that its performance on the overall AUC value is affected significanly from the number of features. While the linear classifiers, irrespective of the number of features, learn with similar tempo, Naive does not follow the same pattern. For example, with 17 features it increases gradually its performance in the first iterations but with 5 fearures it decreases it. In addition, we notice a difference of around 10% in the final AUC value among the cases of 3 and 5 features. This could be explained by the fact that Naive Bayes assumes independence so the combinations of the feautures may affect its ability to accomplish the purchase prediction task especially if we consider the correlations on the experimental dataset. In particular, there are high correlations between the pairs BounceRate-ExitRate and ProductRelated-ProductRelatedDuration. When Naive fitted with 5 features, it actual fitted with these two strong correlative pairs of features and additional with PageValue that was the most informative feature. These strong correlations affected significantly its performance since Naive could only work well with the PageValue feature. On the flipside, when it fitted with only 3 features, there were not features correlated each other since ProductRelated and BounceRate had been

discarded. Therefore, its performance increased in a high degree. In the case that Naive fitted with 7,9 and 17 features, the strong correlations were included but because of the other features, it performed some better compared to the case that only be fitted with 5 features and from these 4 were the most correlative each other. This indication will be confirmed also in the test set but clearly explains the distinctiveness of this classifier.

Figure 16: AUC over iterations of HT classifier with different number of features

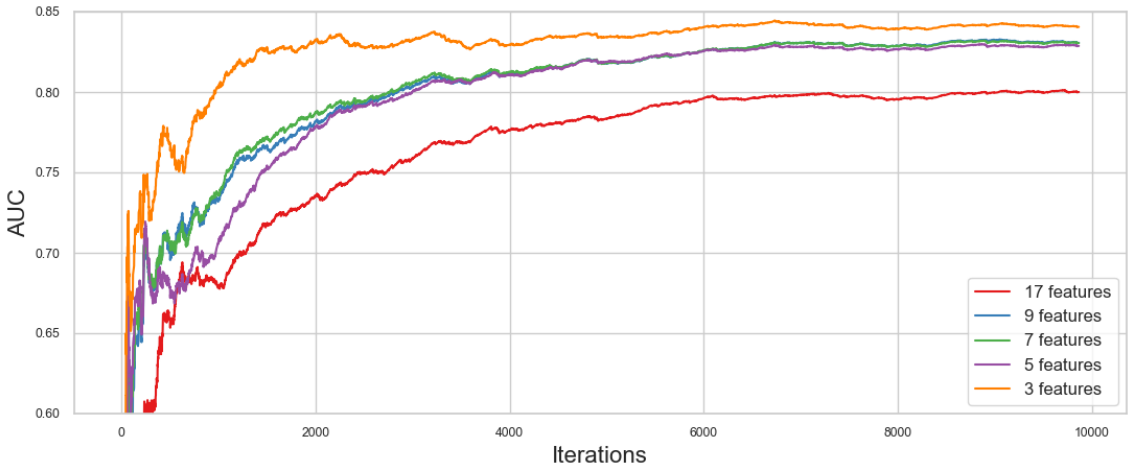


Figure 16 illustrates the progressive of HT in terms of AUC value over iterations. It seems that the feature set with the maximum number of 17 features makes its learning process lower since the AUC value presents fluctuations until 1000 iterations and then until 6000 iterations the classifier stablishes its behavior and that results to the lowest overall performance. On the flip side, when the classifier is fitted with only 3 features, it takes about 2000 iterations to be as much accurate as possible meaning that is learning process is speeded. Among 5,7 and 9 features the learning process is similar. However, in any case, there are some fluctuations on the curves indicating that there is space for improvement.

Figure 17: AUC over iterations of HAT classifier with different number of features

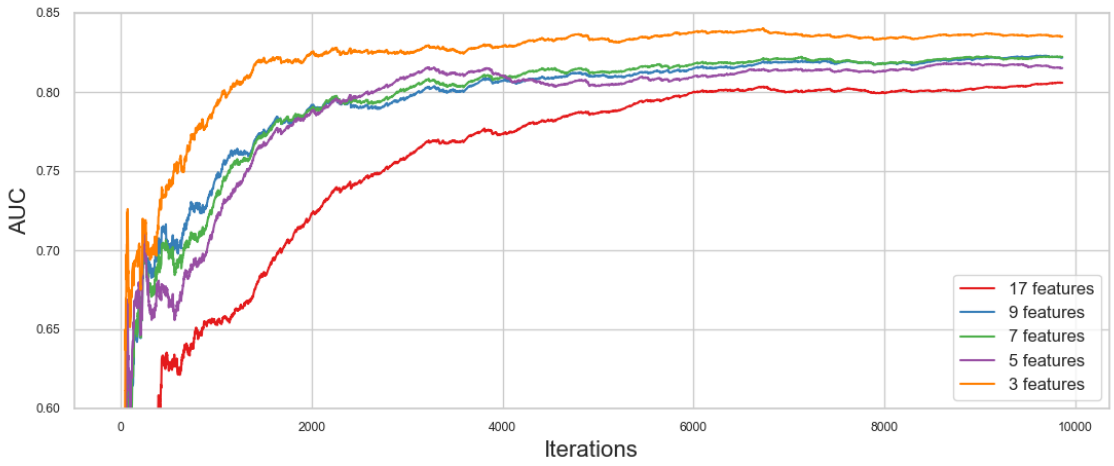
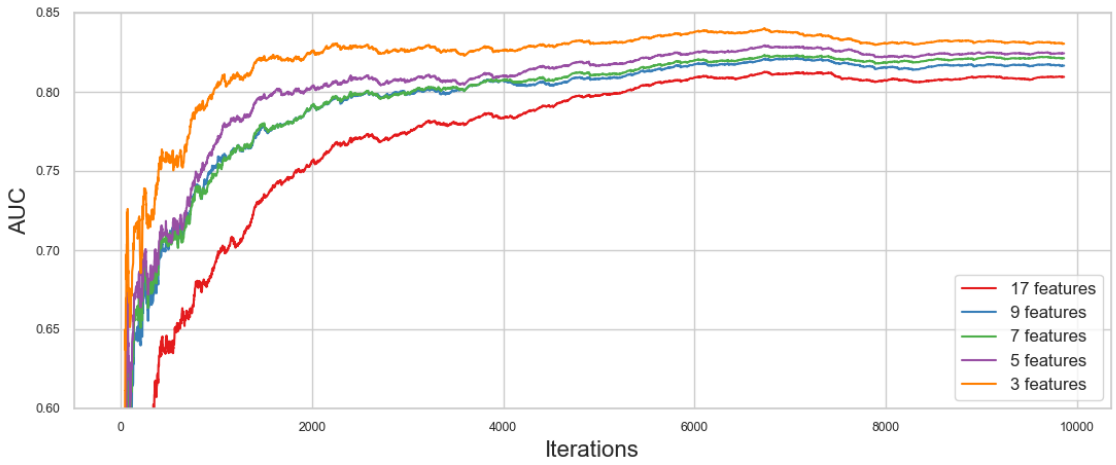


Figure 17 displays the progressive of HAT in terms of AUC value over iterations. Its learning process is almost the same with the HT. It does not make sense since HAT is just an alternative of HT that is adapted accordingly in case of concept drift [42]. However, there are observed less fluctuations that could be the result of ADWIN, the drift detector that is used from HAT to handle the change in data distribution.

Figure 18: AUC over iterations of EFDT classifier with different number of features



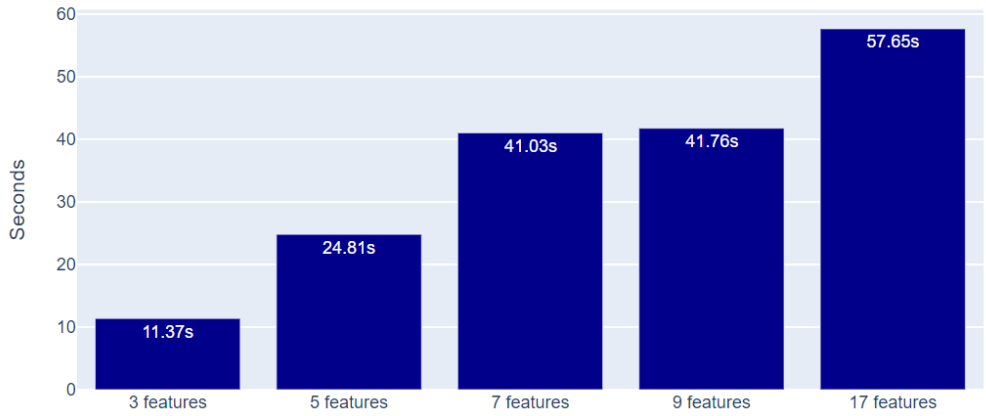
Regarding EFDT (Figure 18) it is notable the fact that regardless the number of features, EFDT does not present a lot of fluctuations in the early iterations compared the other tree-based classifiers, but it improves gradually its performance and continuously learn. This is an indication that EFDT learns faster than the other tree-based classifiers and confirms [56]. However, when it is trained with 3 features its AUC value is consolidates a little faster while with 17 features the procedure gets slower.

6.4.2 Impact of number of features in classifiers performance in training and prediction time

It is worth to comment on the impact of the number of features in training – predict time. Especially, it observed that Naive and Tree-based models are the most time consuming. Linear models required significant short time for such amount of data, so it does not make sense to examine their performance when fitted with different number of features. Also, to point out again that in general the times are short due to the amount of data, but we still get a picture of the impact that dimensionality has on each classifier.

In the following diagrams (Figures,19,20,21,22) it is meaningful to observe that in the majority of cases, the reduction of features significantly improves the time needed by each classifier. This is of course more apparent when we look at EFDT which was observed to need a lot of resources. An important conclusion is that even with only 3 features it takes more time to train EFDT compared to the rest classifiers even when they be fitted with 17 features. The exception is HAT which when uses 3 features takes about the same time as when EFDT with 17 features. This finding highlights the complexity of EFDT and confirms [56] which introduced EFDT and commented on its complexity. It is a trade-off between the efficiency of this classifier and the resources it requires.

Figure 19: Training-prediction time of EFDT



-Figure 20: Training-prediction time of HAT

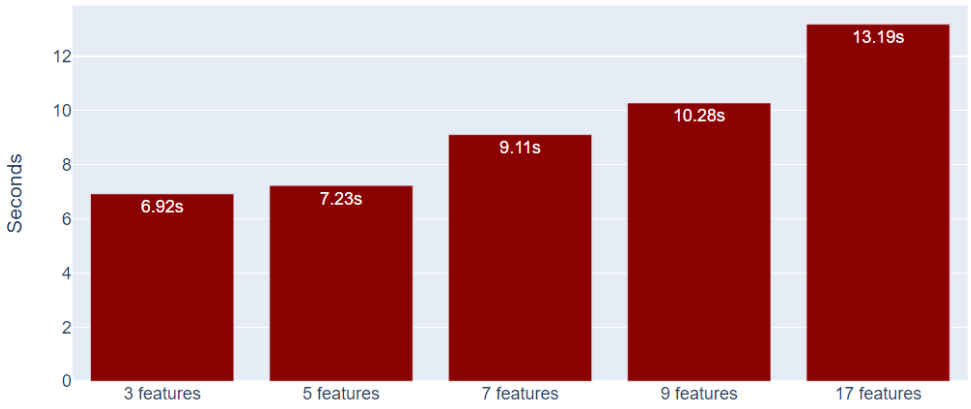


Figure 21: Training-prediction time of HT

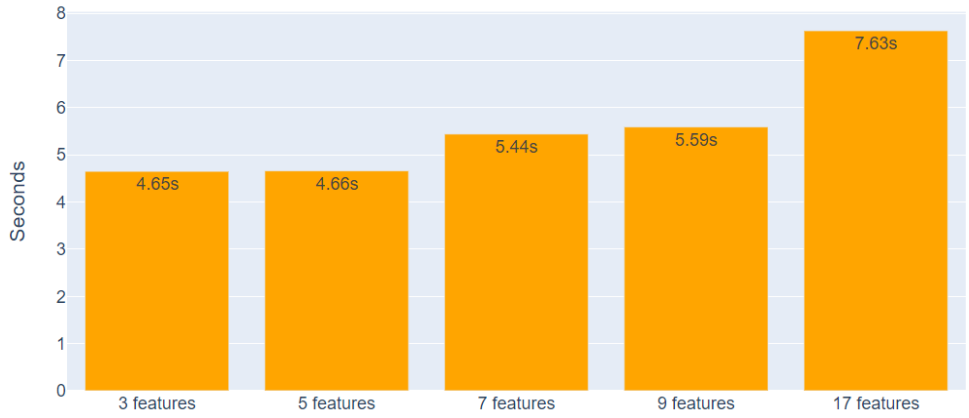
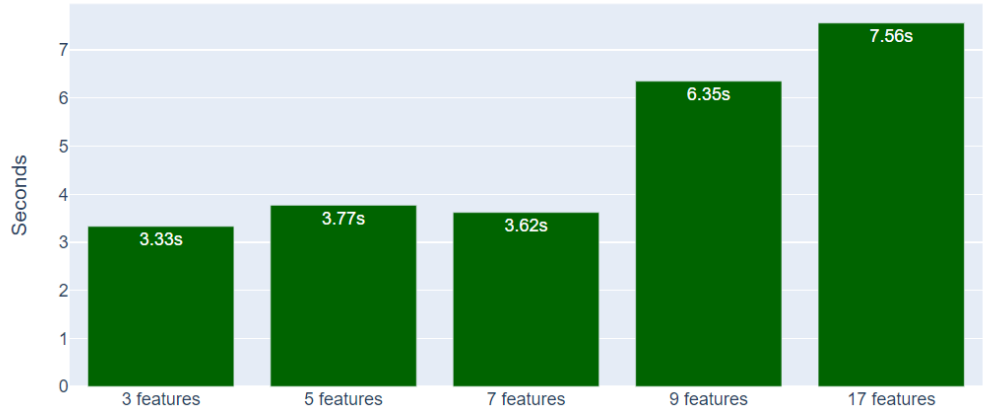


Figure 22: Training-prediction time of Naive Bayes



6.4.3 Impact of number of features in classifiers performance in the test set

After applying feature selection and evaluating the models we can conclude that the model's performance has been improved in terms of AUC for all the models even to a small extent (Figures 23,24). **SGD** and PA seem not to be affected in a large degree from the number of features and achieve around 79%-80% AUC value. The unstable behavior of Naive Classifier which its performance decreased significantly for 5 features but for 3 the classifier is much more efficient, happens due to the correlations that explained earlier.

Tree-based models have larger ability to distinct the two classes compared the other classifiers regardless of the number of features used. HT showed promising results after features reduction but HAT and EFTD fitted with 7 features achieved the highest AUC value (84%).

Considering the time complexity of EFDT, HAT seems to be the optimal classifier to distinct sessions that are going to offer revenue from those who won't. However, SGD and PA could be also candidates for this task in case that a business was interest for a model that requires limited resources.

Figure 23: : AUC after oversampling and hyper-tuning in the test set for PA, SGD and Naive with different number of features

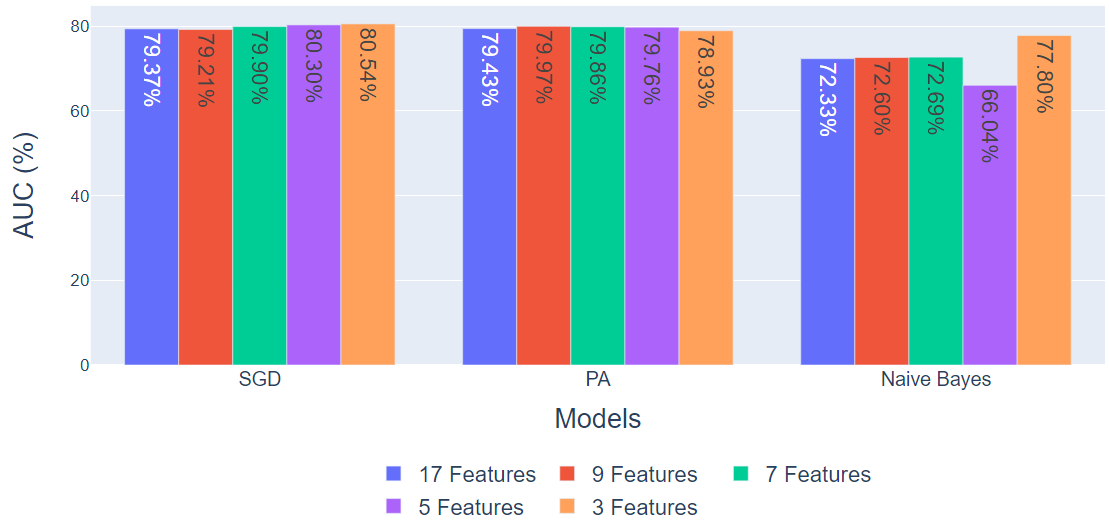
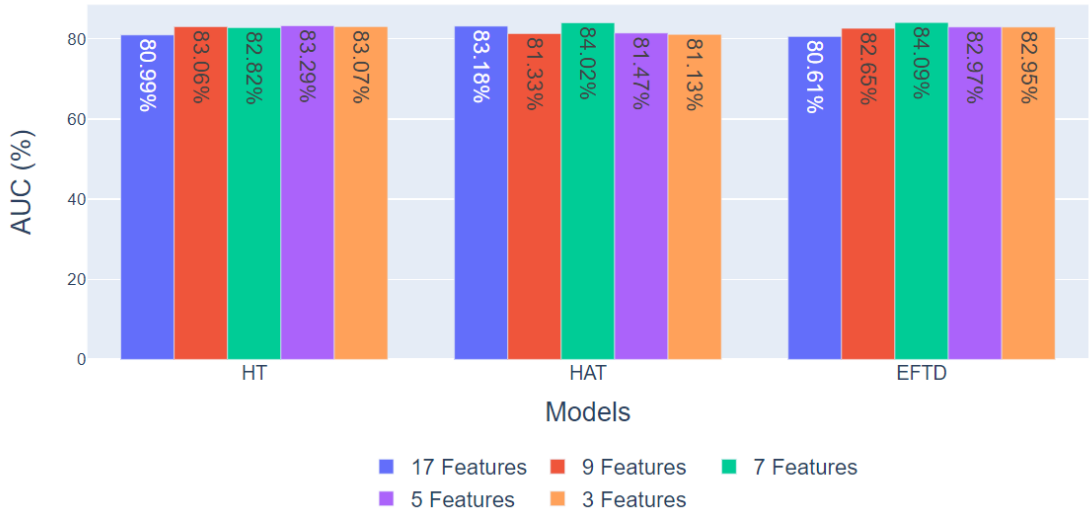


Figure 24: AUC after oversampling and hyper-tuning in the test set for tree-based models with different number of features



Identifying customers who are going to abandon seem to be easier task for all the classfiers since all of them achieve high specificity values independently the number of features except of Naive (Figures 25,26). The latter however shows a peak in its performance in terms of specificity when be trained in the three most important features that mutual information selector suggested and the reason explained earlier. Taking a look into its specificity with 3 and 5 features, the statement mentioned before about correlations becomes even more noticeable. Indeed, its capability to categorizing customers who have not purchase intention is impressive when fitted with only 3 features (specificity 89.01%).

SGD classifier is also powerful for this task in case of be fitted with only 3 features. PA also can determine these customers with high efficiency regardless the number of attributes.

Among tree-based classifiers, all of them have sufficient specificity. In particular, the performance of HT seems not to change significantly independently the dimensionality and it also achieves the highest specificity (89.20%) among all the models. However, when be trained with all the features it has a slightly better metric. Indeed, is the only one that performs better in terms of specificity when all the features are available. The reduction of dimensionality benefited at most HAT classifier to identify the customers with browsing intent. Its specificity is quite better when limited features participate. EFDT suggest also promising results except of the cases of 17 or 7 predictors. The fact

that achieves high scores with limited numbers of features benefit him because the process is quicker, and its complexity reduces.

Figure 25: Specificity after oversampling and hyper-tuning in the test set for PA, SGD and Naive with different number of features

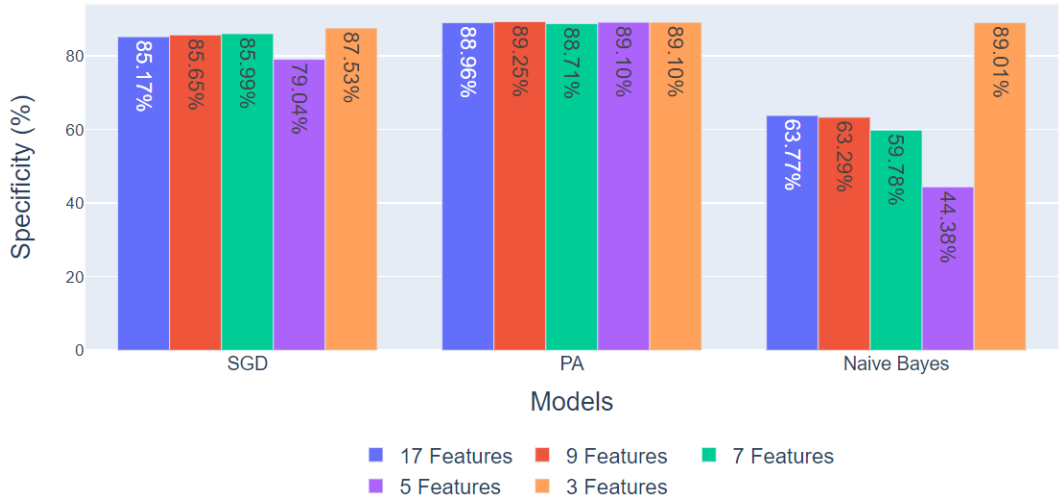
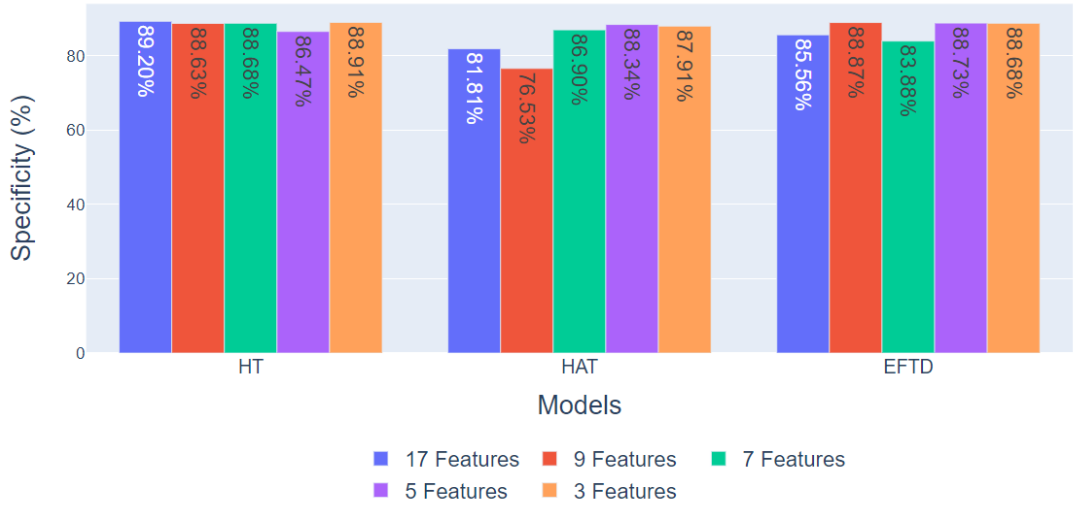


Figure 26: Specificity after oversampling and hyper-tuning in the test set for tree-based models with different number of features



Sensitivity values in general are lower from the specificity one's meaning that «promising» customers are more difficult to be determined in this experiment (Figures 27,28). However, in some cases feature selection process contributed to overcome this obstacle.

Regarding linear classifiers, both are not able to identify these users with high accuracy and especially PA. With respect to Naive, it looks like the correlations affected mainly the specificity rather than the sensitivity of the model since sensitivity is related high.

The higher value achieved when trained incrementally with 5 features. However, in this case its ability to identify the negative class is very poor (specificity=44.48) so it wouldn't be an optimal choice. However, it may could contribute with some combinations of features (17,9,7) only in case that a business was interesting identifying only these kinds of customers. Among tree-based models, HT shows the lower values. HAT increases its ability to accomplish this challenge when fitted with many features (17 or 9). Under these circumstances, the classifier had lower specificity. So, the dimensionality reduction contributed only to its specificity. Finally, EFDT can accomplish it with succeed only in case of 7 features.

Figure 27: Sensitivity after oversampling and hyper-tuning in the test set for PA, SGD and Naive with different number of features

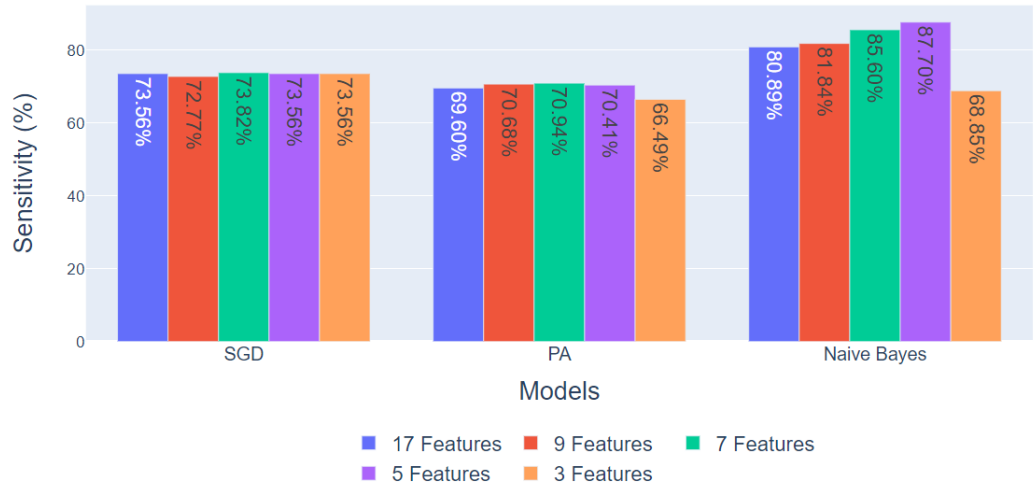
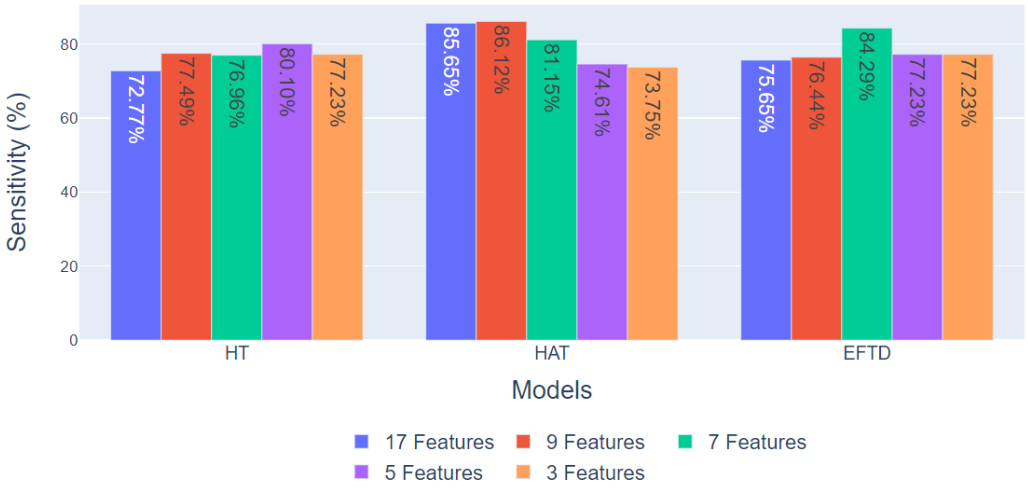


Figure 28: Sensitivity after oversampling and hyper-tuning in the test set for tree-based models with different number of features



7 Conclusions

7.1 Discussion and conclusions

In this research, online machine learning algorithms were applied to predict purchase intention in real time. It's an issue that is becoming more and more important with the continued growth of e-commerce. Businesses constantly want to increase their profit by both keeping their customers or gaining new ones. The goal for each business is to convert as most as possible browsing customers into buyers and simultaneously to be as more as possible benefited from the buyers. The idea for using online machine learning algorithms generated based on their ability to be updated over time without any retraining needed. If we consider the amount of data that are generated in daily basis from the users who visit an e-commerce, handling them efficiency is required. Utilizing these algorithms limited resources are needed and in addition useful insights are extracted in real-time that allows immediately decisions that can contribute to customer journey.

Several online classifiers including tree-based, linear, and probabilistic one's competed with aim to achieve this task. Naive Bayes that is a probabilistic classifier has a peculiarity since while it assumes independence between the features, in the dataset we have strong correlations, but it was chosen to determine its performance in this case and how it is affected.

Sensitivity, specificity, and AUC metrics got into consideration when it came to evaluated them. The experimental dataset was a well-known that has used in several studies aiming purchase intention prediction and has browsing behavior attributes like google analytics metrics.

On technical view, a first observation that occurred is that only Naive Bayes classifier managed to achieve satisfactory results on the imbalanced dataset. This is a sign that even the strong correlations, Naive managed to a certain extent to distinguish customers with purchase intention with those with browsing intention, meaning that in this task, Naive was not so sensitive to overfitting compared to the other classifiers who's its sensitivity was worst or almost the same as a random classifier. However, the oversampling of minority task was vital to achieve better results.

After minority oversampling, hyperparameter tuning for linear classifiers proved essential as it improved their performance more than 10% and highlighted impact of this step.

In addition, when monitoring the AUC value in iterations for the various values of passive aggressiveness and learning rate it was seen that it affects the learning of an algorithm since small values lead to slower but more accurate converge.

Subsequently, all the algorithms were fitted with different feature sets including 17,9,7,5, and 3 features and valuables insights extracted according to the impact of the number of features in training-prediction time, on the learning process and on the evaluation metrics on the test set.

Firstly, monitoring the AUC value of each classifier as it was seeing more and more data, led to useful insights about model learning procedure and the relation with the number of features**.** SGD was the one that its learning process affected in the smallest degree. PA proved that with less features, (3 or 5) it manages to distinct the two classes some earlier compared the cases with more features. In general, some fluctuations were observed in all the classifiers, which means that the overall AUC changed during the real time predictions, and this could indicate some change in the browsing behavior of the consumers. Among tree-based learners, all suggested similar learning procedure. The fewer the features, the faster it captures the patterns of the data. Especially the HAT learns almost the same way as the HT and this is because they have similar architecture. However, less fluctuations were observed in HAT and this can be explained by the advantage of this algorithm to adapt in the change of the data distribution or concept drift as it called. Additionally, EFDT was shown to learn with efficiency from the first iterations and is confirmed to be a "Fast Learner". The peculiarity of Naive Bayes was also shown and proved that strong correlations can affect its performance significantly. Especially in the case that it was fitted with 5 features, and from them, 4 were strong correlated each other («ExitRate-«BounceRate» and «ProductRelated»-«ProductRelatedDuration») the performance decreased to an impressive degree but when only 3 features participated, and strong correlated attributes were absent, Naive's power was proven since his performance increased rapidly.

It is worth mentioning the impact of dimensionality on the required time of each classifier. It proved that the dimensionality reduction, speed the process for all the classifiers. Despite that, EFDT even be fitted with only 3 features, undoubtedly its performance regarding this factor is improving but it requires the same or mote time than the other classifiers fitted with 17 features and that emphasizes its complexity.

HAT and EFDT achieved the highest AUC values (84%) meaning that they are the optimal candidates to distinguish customers that are going to offer revenue to a business from those who won't. This was a result when the algorithms fitted with 7 features. Taking into consideration the complexity of EFDT, and the adaptability of HAT, HAT is considered the best classifier for the purchase intention task if AUC is the only interested metric. However, when in EFDT learning and predicting procedure participated only 3 features, AUC value of 83% observed so if EFDT is the model of interest, the limited features would be much better choice because it would reduce the complexity very significantly with only a minimum impact on AUC value. In addition, HT performed very well achieving an AUC score around 83% when it fitted with 3,5,7, or 9 features. Linear models did not overcome the tree-based one's, but SGD also showed a good performance in terms of AUC value especially when it fitted with 3 and 5 features (81%). Consequently, if limited resources and the complexity of the models is a determining factor, SGD would be a good choice.

In terms of specificity, meaning identifying customers that engage with the products but with browsing intent all the classifiers presented precious results except of Naive Bayes that only when trained with 3 features suggested very promising outcome (specificity 89%). In addition, HAT suggested high specificity only when less than 7 features participated. PA may could be the winner regarding this metric since regardless the dimensionality, achieved specificity around 90%. Surely, a business that would like to focus into attracting users with intent not to buy, has various choices of selecting a model.

In contrast, determining which customers are likely to proceed to a purchase, was more difficult task for the classifiers since only a small ratio of customers belongs to this category. Linear-based models did not manage to accomplish this task with efficiency. The best results suggested from HAT when training with 17 or 9 features (sensitivity around 86%) and Extremely Fast Decision Tree when training with 7 features (sensitivity 84%) showing the dynamics of tree-based models.

To conclude and answer on the research questions, online classifiers suggest promising results and can accomplish the purchase prediction task with quite efficiency but do not overcome the literature that used offline learners in this task. Tree-based models seem to overcome linear and probabilistic models achieving higher evaluation metrics in most of the experiments. However, linear models are the less time consuming. Feature dimensionality has an impact on the model's capability and affects each model in a differ-

ent way. In any case, it contributes on the time complexity and speed learning process and as a result the models are more accurate in real-time predictions even in a small degree. Only SGD's learning procedure did not affect by the number of features. In this experiment, the limitation of the Naive algorithm, which assumes that all features are independent, was clearly seen, and the difference in its performance when there are independencies and when there are none highlighted.

With the usage of online machine learning methods, business can be benefited in a high degree. The nature of these algorithms to learn continuously and be updated according to the consumer behavior in real-time, provides opportunities for immediate data-drive actions aiming to increase conversion rate. Marketing campaigns can be adjusted accordingly considering the needs of the business providing offers, recommendations, and personalized experience avoiding target browsers like buyers and vice versa.

7.2 Limations and suggestions for improvement

This study has limitations; however, it is a basis to further study the problem of purchase intention using online machine learning algorithms that suit the ever-changing needs of customers. The ability of these algorithms to always stay updated with new data can be a very powerful tool for this issue. A limitation is that the algorithms were implemented only on a relatively small dataset. It is therefore suggested these algorithms to be applied with more data sets suitable for purchase intention prediction.

In addition, another limitation is that the feature selection was made offline which is not feasible in real scenarios as data come continuously and the distribution may change over time. In the concept of streaming data, it is not guaranteed that the same features will provide the same information over time. So, to improve this study, feature selection should be done dynamically, and the algorithms should gradually learn each time from the most informative features. Like that, due to the small amount of data, the hyper tuning was done assuming that all the data is available. This is not possible in real conditions, so for even better results we recommend hyper tuning in the online setting.

The ability of online algorithms to adapt to concept drift was not optimally utilized in the present study. Only HAT was used which by its nature adapts according to the distribution of the data. The inclusion of drift detection algorithms is suggested. By lever-

aging these methods, and identifying the change in consumer behavior, the algorithms will be able to adapt better.

Furthermore, an additional suggestion for improvement is to use these algorithms with windows sizes, even in the case of larger datasets, so to learn from the most recent data and examine if this technique is beneficial or not for this task. As well, the evaluation of the online algorithms in this research was done by monitoring the AUC considering the data that the classifier has seen so far. In a larger volume of data, evaluating the algorithms with prequential metrics could be more efficient so that concept drift can be detected easier anytime.

A concluding strategy for optimization, contains the usage of online clustering algorithms with aim to categorize the customers into clusters. It is very important for every business to know its customers as this way it can more easily adjust its strategy accordingly. By using online clustering algorithms, the business will be able to draw useful conclusions and know at every moment how the customers behave. Therefore, the combination of clustering in real time with purchase intention prediction in real time is proposed so that a company can make the most of the dynamics of online machine learning methods in to continuously improve the experience of its customers and thus increase its profits.

To conclude, this work implemented utilizing online algorithms using simple techniques of data streams. However, it motivates many variations that could optimize the increase of a business's conversion rate and the targeting marketing campaigns.

Bibliography

[1] Statista. (2021). Worldwide retail e-commerce sales from 2014 to 2026 (in billion U.S. dollars). https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[2] Guo, L., Hua, L., Jia, R., Zhao, B., Wang, X., & Cui, B. (2019, July). Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1984-1992).

[3] Song, J. D. (2019). A study on online shopping cart abandonment: a product category perspective. Journal of Internet Commerce, 18(4), 337-368.

[4] Indiani, N. L. P., & Fahik, G. A. (2020). Conversion of online purchase intention into actual purchase: the moderating role of transaction security and convenience. Business: Theory and Practice, 21(1), 18-29.

[5] Tang, L., Wang, X., & Kim, E. (2022). Predicting Conversion Rates in Online Hotel Bookings with Customer Reviews. Journal of Theoretical and Applied Electronic Commerce Research, 17(4), 1264-1278.

[6] Tong, T., Xu, X., Yan, N., & Xu, J. (2022). Impact of different platform promotions on online sales and conversion rate: The role of business model and product line length. Decision Support Systems, 156, 113746.

[7] Surendro, K. (2019, March). Predictive analytics for predicting customer behav-ior. In 2019 International Conference of Artificial Intelligence and Information Tech-nology (ICAIIT) (pp. 230-233). IEEE.

[8] Iqbal, M. (2022). Machine Learning Applications in E-Commerce. Organization, Business and Management, 65.

[9] Kumari, R., & Srivastava, S. K. (2017). Machine learning: A review on binary classification. International Journal of Computer Applications, 160(7).

[10] Burlutskiy, N., Petridis, M., Fish, A., Chernov, A., & Ali, N. (2016). An investigation on online versus batch learning in predicting user behaviour. In Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV 33 (pp. 135-149). Springer International Publishing.

[11] Putatunda, S. (2021). Practical Machine Learning for Streaming Data with Py-thon.

[12]Statista. (2021). Web visits to Amazon.com from October 2019 to July 2021 https://www.statista.com/statistics/623566/web-visits-to-amazoncom/

[13]Sheng, L. K., & Wah, T. Y. (2011). A comparative study of data mining techniques in predicting consumers' credit card risk in banks. African Journal of Business Management, 5(20), 8307.

[14]Amezzane, I., Fakhri, Y., Aroussi, M. E., & Bakhouya, M. (2019). Comparative study of batch and stream learning for online smartphone-based human activity recognition. In Innovations in Smart Cities Applications Edition 2: The Proceedings of the Third International Conference on Smart City Applications (pp. 557-571). Springer International Publishing.

[15]Corrêa, D. G., Enembreck, F., & Silla, C. N. (2017, May). An investigation of the hoeffding adaptive tree for the problem of network intrusion detection. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 4065-4072). IEEE.

[16] Martindale, N., Ismail, M., & Talbert, D. A. (2020). Ensemble-based online machine learning algorithms for network intrusion detection systems using streaming da-ta. Information, 11(6), 315.

[17]Data, M., & Aritsugi, M. (2022, July). AB-HT: An Ensemble Incremental Learning Algorithm for Network Intrusion Detection Systems. In 2022 International Conference on Data Science and Its Applications (ICoDSA) (pp. 47-52). IEEE.

[18]Nagashri, K., & Sangeetha, J. (2021). Fake news detection using passive-aggressive classifier and other machine learning algorithms. In Advances in Compu-ting and Network Communications: Proceedings of CoCoNet 2020, Volume 2 (pp. 221-233). Springer Singapore.

[19] Shahraki, A., Abbasi, M., Taherkordi, A., & Jurcut, A. D. (2022). 1. Computer Networks, 207, 108836.

[20]Oshima, K., Yamamoto, D., Yumoto, A., Kim, S. J., Ito, Y., & Hasegawa, M. (2022). Online machine learning algorithms to optimize performances of complex wireless communication systems. Mathematical Biosciences and Engineering, 19(2), 2056-2094.

[21]Yao, J., & Ansari, N. (2019). Task allocation in fog-aided mobile IoT by Lyapun-ov online reinforcement learning. IEEE Transactions on Green Communications and Networking, 4(2), 556-565.

[22] Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N., & Dellnitz, M. (2020). Deep model predictive flow control with limited sensor data and online learning. Theoretical and computational fluid dynamics, 34, 577-591.

[23]Ade, R., & Deshmukh, P. R. (2014). Instance-based vs batch-based incremental learning approach for students classification. International Journal of Computer Applications, 106(3).

[24] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

[25] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), 160.

[26]Vaidya, N., & Khachane, A. R. (2017, July). Recommender systems-the need of the ecommerce ERA. In 2017 International Conference on Computing Methodologies and Communication (ICCMC) (pp. 100-104). IEEE.

[27] Al-Mashraie, M., Chung, S. H., & Jeon, H. W. (2020). Customer switching behavior analysis in the telecommunication industry via push-pull-mooring framework: A machine learning approach. Computers & Industrial Engineering, 144, 106476.

[28]Esmeli, R., Bader-El-Den, M., & Abdullahi, H. (2021). Towards early purchase intention prediction in online session based retailing systems. Electronic Mar-kets, 31(3), 697-715.

[29] Carbonara, L., & Borrowman, A. (1998). A comparison of batch and incremental supervised learning algorithms. In Principles of Data Mining and Knowledge Discovery: Second European Symposium, PKDD'98 Nantes, France, September 23–26, 1998 Proceedings 2 (pp. 264-272). Springer Berlin Heidelberg.

[30]Losing, V., Hammer, B., & Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing, 275, 1261-1274.

[31]Hoi, S. C., Sahoo, D., Lu, J., & Zhao, P. (2021). Online learning: A comprehensive survey. Neurocomputing, 459, 249-289.

[32] Lohrasbinasab, I., Shahraki, A., Taherkordi, A., & Delia Jurcut, A. (2022). From statistical-to machine learning-based network traffic prediction. Transactions on Emerging Telecommunications Technologies, 33(4), e4394.

[33] Hoi, S.C., Wang, J., Zhao, P.: Libol: a library for online learning algorithms. J. Mach. Learn. Res. 15, 495–499 (2014)

[34] Read, J., Bifet, A., Pfahringer, B., & Holmes, G. (2012). Batch-incremental versus instance-incremental learning in dynamic and evolving data. In Advances in Intelligent Data Analysis XI: 11th International Symposium, IDA 2012, Helsinki, Finland, October 25-27, 2012. Proceedings 11 (pp. 313-323). Springer Berlin Heidelberg.

[35] Wang, S., Minku, L. L., & Yao, X. (2013, April). A learning framework for online class imbalance learning. In 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL) (pp. 36-45). IEEE.

[36] Madhavan, S., & Kumar, N. (2021). Incremental methods in face recognition: a survey. Artificial Intelligence Review, 54(1), 253-303.

[37] Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., ... & Pothuhera, D. (2019). Online incremental machine learning platform for big data-driven smart traffic management. IEEE Transactions on Intelligent Transportation Systems, 20(12), 4679-4690.

[38] Bisong, E., & Bisong, E. (2019). Batch vs. online learning. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, 199-201.

[39] Ade, R., & Deshmukh, P. R. (2014). Instance-based vs batch-based incremental learning approach for students classification. International Journal of Computer Applications, 106(3).

[40] Hong, X., Guan, S. U., Man, K. L., & Wong, P. W. (2020). Lifelong machine learning architecture for classification. Symmetry, 12(5), 852.

[41] Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining data streams: a review. ACM Sigmod Record, 34(2), 18-26.

[42] Putatunda, S. (2021). Practical Machine Learning for Streaming Data with Python. Apress.

[43] Zhang, S. S., Liu, J. W., & Zuo, X. (2021). Adaptive online incremental learning for evolving data streams. Applied Soft Computing, 105, 107255.

[44] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. IEEE transactions on knowledge and data engineering, 31(12), 2346-2363.

[45] Kholghi, M., & Keyvanpour, M. (2011). An analytical framework for data stream mining techniques based on challenges and requirements. arXiv preprint arXiv:1105.1950.

[46] Adewole, K. S., Salau-Ibrahim, T. T., Imoize, A. L., Oladipo, I. D., AbdulRaheem, M., Awotunde, J. B., ... & Aro, T. O. (2022). Empirical Analysis of Data Streaming and Batch Learning Models for Network Intrusion Detection. Electronics, 11(19), 3109.

[47] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

[48]Patel, B. R., & Rana, K. K. (2014). A survey on decision tree algorithm for classification. International Journal of Engineering Development and Research, 2(1), 1-5.

[49]Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering, 6(10), 74-78.

[50] Rajaguru, H., & SR, S. C. (2019). Analysis of decision tree and k-nearest neighbor algorithm in the classification of breast cancer. Asian Pacific journal of cancer prevention: APJCP, 20(12), 3777.

[51] Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

[52] Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 71-80).

[53] Bifet, A., & Gavalda, R. (2009). Adaptive learning from evolving data streams. In Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31-September 2, 2009. Proceedings 8 (pp. 249-260). Springer Berlin Heidelberg.

[54] Putatunda, S. (2021). Practical Machine Learning for Streaming Data with Py-thon. Apress.

[55] Hulten, G., Spencer, L., & Domingos, P. (2001, August). Mining time-changing data streams. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 97-106).

[56] Manapragada, C., Webb, G. I., & Salehi, M. (2018, July). Extremely fast decision tree. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1953-1962).

[57]Hue, C., Boullé, M., & Lemaire, V. (2017). Online learning of a weighted selec-tive naive Bayes classifier with non-convex optimization. Advances in Knowledge Discovery and Management: Volume 6, 3-17.

[58] Godec, M., Leistner, C., Saffari, A., & Bischof, H. (2010, August). On-line random naive bayes for tracking. In 2010 20th International Conference on Pattern Recognition (pp. 3545-3548). IEEE.

[59] Gumus, F., Sakar, C. O., Erdem, Z., & Kursun, O. (2014, August). Online Naive Bayes classification for network intrusion detection. In 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014) (pp. 670-674). IEEE.

[60] Prinzie, A., & Van den Poel, D. (2007). Random multiclass classification: Generalizing random forests to random mnl and random nb. In Database and Expert Systems Applications: 18th International Conference, DEXA 2007, Regensburg, Germany, September 3-7, 2007. Proceedings 18 (pp. 349-358). Springer Berlin Heidelberg.

[61]Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29, 103-130.

[62]Hand, D. J., & Yu, K. (2001). Idiot's Bayes—not so stupid after all?. International statistical review, 69(3), 385-398.

[63] Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006, July). Spam filtering with naive bayes-which naive bayes?. In CEAS (Vol. 17, pp. 28-69).

[64]Ting, S. L., Ip, W. H., & Tsang, A. H. (2011). Is Naive Bayes a good classifier for document classification. International Journal of Software Engineering and Its Applications, 5(3), 37-46.

[65] Amezzane, I., Fakhri, Y., Aroussi, M. E., & Bakhouya, M. (2019). Comparative study of batch and stream learning for online smartphone-based human activity recognition. In Innovations in Smart Cities Applications Edition 2: The Proceedings of the

Third International Conference on Smart City Applications (pp. 557-571). Springer International Publishing.

[66] Salperwyck, C., & Lemaire, V. (2011, July). Learning with few examples: An empirical study on leading classifiers. In The 2011 international joint conference on neural networks (pp. 1010-1019). IEEE.

[67] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive aggressive algorithms.

[68] Lu, J., Zhao, P., & Hoi, S. (2015, February). Online passive aggressive active learning and its applications. In Asian Conference on Machine Learning (pp. 266-282). PMLR.

[69] Gupta, S., & Meel, P. (2021). Fake news detection using passive-aggressive classifier. In Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 (pp. 155-164). Springer Singapore.

[70] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers (pp. 177-186). Physica-Verlag HD.

[71] Zhang, T. (2004, July). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of the twenty-first international conference on Machine learning (p. 116).

[72] Richtárik, P., & Takáč, M. (2016). Parallel coordinate descent methods for big data optimization. Mathematical Programming, 156, 433-484.

[73] Raschka, S. (2014). An overview of general performance metrics of binary classifier systems. arXiv preprint arXiv:1410.5330.

[74] Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: binary classification for heterogeneous datasets. SMU Data Science Review, 1(3), 9

[75] Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: a better measure than accuracy in comparing learning algorithms. In Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003, Halifax, Canada, June 11–13, 2003, Proceedings 16 (pp. 329-341). Springer Berlin Heidelberg.

[76] Wang, S., Li, D., Petrick, N., Sahiner, B., Linguraru, M. G., & Summers, R. M. (2015). Optimizing area under the ROC curve using semi-supervised learning. Pattern
[77] Vujović, Ž. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599-606.

[78] Bifet, A., de Francisci Morales, G., Read, J., Holmes, G., & Pfahringer, B. (2015, August). Efficient online evaluation of big data stream classifiers. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 59-68).

[79]Brzezinski, D., & Stefanowski, J. (2015). Prequential AUC for classifier evaluation and drift detection in evolving data streams. In New Frontiers in Mining Complex Patterns: Third International Workshop, NFMCP 2014, Held in Conjunction with ECML-PKDD 2014, Nancy, France, September 19, 2014, Revised Selected Papers 3 (pp. 87-101). Springer International Publishing.

[80] Mulinka, P., & Casas, P. (2018, August). Stream-based machine learning for network security and anomaly detection. In Proceedings of the 2018 workshop on big data analytics and machine learning for data communication networks (pp. 1-7).

[81] Lichtenwalter R, Chawla NV (2009) Adaptive methods for classification in arbitrarily imbalanced and drifting data streams. In: PAKDD Workshops, Lecture Notes in Computer Science, vol 5669, pp 53–75

[82] Ditzler G, Polikar R (2013) Incremental learning of concept drift from streaming imbalanced data. IEEE Trans Knowl Data Eng 25(10):2283–2301

[83] Bouckaert RR (2006) Efficient AUC learning curve calculation. In: Proceedings of Australian conference on artificial intelligence. Lecture notes in computer science, vol 4304, pp 181–191

[84] Brzezinski, D., & Stefanowski, J. (2017). Prequential AUC: properties of the ar-ea under the ROC curve for data streams with concept drift. Knowledge and Infor-mation Systems, 52, 531-562.

[85] Esmeli, R., Bader-El-Den, M., & Abdullahi, H. (2021). Towards early purchase intention prediction in online session based retailing systems. Electronic Mar-kets, 31(3), 697-715.

[86] Diamantaras, K. I., Salampasis, M., Katsalis, A., & Christantonis, K. (2021). Predicting Shopping Intent of e-Commerce Users using LSTM Recurrent Neural Networks. In DATA (pp. 252-259).

[87] Toth, A., Tan, L., Di Fabbrizio, G., & Datta, A. (2017, August). Predicting Shopping Behavior with Mixture of RNNs. In ecom@ sigir.

[88] Suchacka, G., Skolimowska-Kulig, M., & Potempa, A. (2015). Classification Of E-Customer Sessions Based On Support Vector Machine. ECMS, 15, 594-600

[89]Ahsain, S., & Kbir, M. A. (2022). Predicting the client's purchasing intention us-ing Machine Learning models. In E3S Web of Conferences (Vol. 351, p. 01070). EDP Sciences.

[90] Kabir, Md Rayhan & Ashraf, Faisal Bin & Ajwad, Rasif. (2019). Analysis of Different Predicting Model for Online Shoppers' Purchase Intention from Empirical Data. 1-6. 10.1109/ICCIT48885.2019.9038521.

[91] Parihar, V., & Yadav, S. (2022). Comparative Analysis of Different Machine Learning Algorithms to Predict Online Shoppers' Behaviour. International Journal of Advanced Networking and Applications, 13(6), 5169-5182.

[92] Sakar, C. O., Polat, S. O., Katircioglu, M., & Kastro, Y. (2019). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Computing and Applications, 31, 6893-6908.

[93] Esmeli, R., Bader-El-Den, M., & Abdullahi, H. (2022). An analyses of the effect of using contextual and loyalty features on early purchase prediction of shoppers in e-commerce domain. Journal of Business Research, 147, 420-434.

[94] Chaudhuri, N., Gupta, G., Vamsi, V., & Bose, I. (2021). On the platform but will they buy? Predicting customers' purchase behavior using deep learning. Decision Support Systems, 149, 113622.

[95] Rausch, T. M., Derra, N. D., & Wolf, L. (2022). Predicting online shopping cart abandonment with machine learning approaches. International Journal of Market Research, 64(1), 89-112.

[96] Rifat, M. R. I., Amin, M. N., Munna, M. H., & Al Imran, A. (2022, September). An End-to-end Machine Learning System for Mitigating Checkout Abandonment in E-Commerce. In 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 129-132). IEEE

[97] Masrani, A., Shukla, M., & Makadiya, K. (2020, August). Empirical Analysis of Classification Algorithms in Data Stream Mining. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 1 (pp. 657-669). Singapore: Springer Singapore.

[98]Nagashri, K., & Sangeetha, J. (2021). Fake news detection using passive-aggressive classifier and other machine learning algorithms. In Advances in Compu-ting and Network Communications: Proceedings of CoCoNet 2020, Volume 2 (pp. 221-233). Springer Singapore.

[99] Data, M., & Aritsugi, M. (2022, July). AB-HT: An Ensemble Incremental Learning Algorithm for Network Intrusion Detection Systems. In 2022 International Conference on Data Science and Its Applications (ICoDSA) (pp. 47-52). IEEE.

[100]Adewole, K. S., Salau-Ibrahim, T. T., Imoize, A. L., Oladipo, I. D., AbdulRaheem, M., Awotunde, J. B., ... & Aro, T. O. (2022). Empirical Analysis of Data Streaming and Batch Learning Models for Network Intrusion Detection. Electronics, 11(19), 3109.

[101] Burlutskiy, N., Petridis, M., Fish, A., Chernov, A., & Ali, N. (2016). An investigation on online versus batch learning in predicting user behaviour. In Research and De-velopment in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV 33 (pp. 135-149). Springer International Publishing.

[102] Sheng, L. K., & Wah, T. Y. (2011). A comparative study of data mining techniques in predicting consumers' credit card risk in banks. African Journal of Business Management, 5(20), 8307.

[103] Amezzane, I., Fakhri, Y., Aroussi, M. E., & Bakhouya, M. (2019). Comparative study of batch and stream learning for online smartphone-based human activity recognition. In Innovations in Smart Cities Applications Edition 2: The Proceedings of the Third International Conference on Smart City Applications (pp. 557-571). Springer International Publishing.

[104] Read, J., Bifet, A., Pfahringer, B., & Holmes, G. (2012). Batch-incremental versus instance-incremental learning in dynamic and evolving data. In Advances in Intelligent Data Analysis XI: 11th International Symposium, IDA 2012, Helsinki, Finland, Octo-ber 25-27, 2012. Proceedings 11 (pp. 313-323). Springer Berlin Heidelberg.

[105] Gumus, F., Sakar, C. O., Erdem, Z., & Kursun, O. (2014, August). Online Naive Bayes classification for network intrusion detection. In 2014 IEEE/ACM International

Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014) (pp. 670-674). IEEE.

[106] Martindale, N., Ismail, M., & Talbert, D. A. (2020). Ensemble-based online machine learning algorithms for network intrusion detection systems using streaming da-ta. Information, 11(6), 315.

[107] Corrêa, D. G., Enembreck, F., & Silla, C. N. (2017, May). An investigation of the hoeffding adaptive tree for the problem of network intrusion detection. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 4065-4072). IEEE.

[108] Online Shoppers Purchasing Intention Dataset. UCI Machine Learning Repository.
https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset

[109] Frazier, A., Maloku, F., Li, X., Chen, Y., Jung, Y., & Zohuri, B. (2022). Data Analysis of Online Shopper's Purchasing Intention Machine Learning for Prediction Analytics. Journal of Economics & Management Research. SRC/JESMR-191. DOI: doi. org/10.47363/JESMR/2022 (3), 162, 2-8.

[110] Prayogo, R. D., & Karimah, S. A. (2021, September). Feature Selection and Adap-tive Synthetic Sampling Approach for Optimizing Online Shopper Purchase Intent Prediction. In 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) (pp. 1-5). IEEE.

[111] Güzel, B. E. K., & Devrim, Ü. N. A. Y. Predicting Purchase Interest of Online Shoppers Using Boosting Algorithms. Natural and Applied Sciences Journal, 4(2), 1-15.

[112] Siddik, M. A. B., Mazumder, M. M. I., Alam, R., & Khan, M. (2021, April). Performance Comparison Between Dimension Reduction and Feature Selection Approaches for Data Classification. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 893-898). IEEE.

[113] Agustyaningrum, C. I., Haris, M., Aryanti, R., & Misriati, T. (2021). Online shopper intention analysis using conventional machine learning and deep neural network classification algorithm. Jurnal Penelitian Pos dan Informatika, 11(1), 89-1

[114] Prinzie, A., & Van den Poel, D. (2007). Random multiclass classification: Generalizing random forests to random mnl and random nb. In Database and Expert Systems Applications: 18th International Conference, DEXA 2007, Regensburg, Germany, September 3-7, 2007. Proceedings 18 (pp. 349-358). Springer Berlin Heidelberg.

[115] Poulos, M., Korfiatis, N., & Papavlassopoulos, S. (2020). Assessing stationarity in web analytics: A study of bounce rates. Expert Systems, 37(3), e12502.

[116] Ireland, T. N. (2019). Google Analytics basics.

[117] Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., ... & Bifet, A. (2021). River: machine learning for streaming data in python. The Journal of Machine Learning Research, 22(1), 4945-4952.

[118] River, Imbalanced learning, https://rivermlxyz/dev/examples/imbalancedlearning/

[119] River, StandardScaler, https://riverml.xyz/dev/api/preprocessing/StandardScaler/ .

[120] Veloso, B., Gama, J., & Malheiro, B. (2018). Self hyper-parameter tuning for data streams. In Discovery Science: 21st International Conference, DS 2018, Limassol, Cyprus, October 29–31, 2018, Proceedings 21 (pp. 241-255). Springer International Publishing.