The Potential of Deep Learning in Marketing:

Insights from Predicting Conversion with Deep Learning

Name: Rutger Ruizendaal
Student Number: s1225898
Date: 08-09-2017
Supervisors UT: Dr. S.A. de Vries & Dr. E. Constantinides
Supervisors StudyPortals: T. van Vugt & T. Farzami
Study: Communication Studies
Specialization: Marketing Communication

Acknowledgements

The delivery of this master thesis marks the end of my academic studies at the University of Twente. After a bachelor and two master studies my time as a student has come to an end. This master thesis was written in conjunction with an internship at StudyPortals, the global study choice platform. Over the course of the past six months I finished my internship and worked on this master thesis with a lot of passion.

First, I would like to thank StudyPortals for providing me with the opportunity to do this internship. Throughout the internship I was given the freedom and trust to start new data projects that were originally not in the scope of the internship. Additionally, I was given the freedom and materials to implement deep learning algorithms. My focus has been on increasing data quality and automation for which I also built a data analysis application for the Analytics & Consulting Team at StudyPortals. Specifically I would like to thank Thijs van Vugt for his guidance and Tara Farzami for her technical expertise. Next, I would like to thank my supervisors at the University of Twente. I am aware that this is not a typical research for Marketing Communication and I am thankful that my supervisors gave me the freedom to implement this research. They took a chance by supervising me in this research and I believe it paid off. I also want to thank them for providing useful comments and feedback throughout the writing of this master thesis. Additionally, I would like to thank Robert Muster for taking the time to provide feedback as well. His feedback especially helped in restructuring and clarifying the methodology chapter.

Both this master thesis and internship have played a crucial role in my development towards becoming a data scientist. Where my first master thesis acted as an introduction into the field, this master thesis allowed me to dive a lot deeper and apply the machine learning algorithms I had learned about. Additionally, it gave me the opportunity to apply deep learning and learn about the practical implications of running these computationally heavy models. I am very excited to be starting my next challenge as Technical Data Scientist at MIcompany in October. Finally, I would like to thank my family and friends for always supporting me during these busy and sometimes difficult times.

I hope you enjoy reading this master thesis.

Rutger Ruizendaal

Enschede, 2017

Abstract

Jordan and Mitchell (2015) and Najafabadi et al. (2015) have discussed the high potential of deep learning in marketing. At the same time, the hype surrounding deep learning has been exponentially growing and is at an all-time high. However, there are few empirical studies researching applications of deep learning in marketing. This study tries to gain an understanding of the value of deep learning in predicting conversion. In order to fully understand the strengths and weaknesses of deep learning models they are also compared with traditional machine learning models. Specifically, this study attempts to capture the value of deep learning models for predicting conversion.

The dataset for this research has been collected at StudyPortals, the global study choice platform. The dataset consists of click-stream data containing over 56 million events. The dataset has been balanced to contain behaviour from over 36.000 converting users and over 36.000 non-converting users in the period 25-04-2017 till 25-05-2017. When comparing the traditional machine learning models the dataset has been pre-processed (normalization, one-hot encoding) in the way that each specific model requires. For the deep learning models the data has been organized as a sequence. All models have been compared on various metrics including accuracy, precision, recall, f1-score, Logloss, prediction time, training time and the transparency of the model. Hereby the predictive quality and the practical usability of the models gets tested. All Models have been trained on a training set of the data and validated against a test set.

Results show that there are various advantages and disadvantages to using deep learning models when predicting conversion. The main disadvantages are: deep learning models are essentially black-box models, deep learning models require a lot of data to find complex patterns and deep learning models are computationally expensive and time-consuming to train and tune. The main advantages are: deep learning can capture sequential relationships in data, because of the hidden layers deep learning models can learn complex and non-linear functions and the deep learning models showed much better predictive accuracy than the traditional machine learning models. The results indicate that when dealing with tabular data it is advisable to use ensemble models like Random Forest and Gradient Boosted Trees. When the data has a sequential aspect a deep learning model like a Recurrent Neural Network with Lost-Short Term Memory can provide good predictions. The value of deep learning is mainly found in its ability to capture complex patterns in the data which then allows it to make better predictions than traditional machine learning models. The findings of this study are not limited to predicting conversion, but can be generalized towards other marketing cases like churn prediction.

Keywords = deep learning, machine learning, conversion, marketing, predictive modelling

Table of Contents

1. Introduction

Self-driving cars, Google Translate and smart speakers are all powered by deep learning. In recent years deep learning has been at the forefront of many breakthroughs in image recognition, speech recognition and natural language processing. Deep learning focuses on the use of artificial neural networks with multiple hidden layers that are inspired by the human brain. Literature on deep learning often discusses its potential in various areas like finance, education and marketing (Jordan & Mitchell, 2015; Najafabadi et al., 2015). However, in marketing research there are few empirical studies researching the value of deep learning in marketing contexts. Consequently, there is little known about practical applications of deep learning in marketing. This research assists in filling that gap by focusing on a specific marketing problem and exploring the value of deep learning models in marketing. Early research in this field has explored deep learning in predicting the next viewed product category on an e-commerce website (Tamhane, Arora & Warrier, 2017) and in predicting students' next action in a MOOC (Tang, Peterson & Pardos, 2016). Therefore, this study focuses on a different marketing case: predicting conversion. In e-marketing, conversion occurs when a visitor of a website becomes a paying customer. In order to assess the performance of deep learning models it is important to compare their performance with other prediction models as well. Deep learning is a subset of machine learning, which is a broader field focused on the ability of algorithms to learn from big amounts of data. These other machine learning models who are not part of deep learning are in this research referred to as 'traditional machine learning models'. By comparing both types of models on various metrics a more complete overview of the value of deep learning for marketing can be presented. At the end of this research we argue that these findings on predicting conversion can be generalized to other prediction problems in marketing such as prediction churn, purchases and click-through rates.

The predictors of conversion have been studied before any deep learning hype existed. Previous research has found positive relationships between the number of page views by a user and the likelihood to purchase (Bellman, Lohse & Johnson, 1999), as well as a positive relationship between session duration and user conversion (Lin, Hu, Sheng & Lee, 2010). However, Goldstein, Oestreicher-Singer and Barzilay (2017) show that more complex concepts like search diversity also affect the probability of conversion per user. Therefore, it is crucial to include other measures than merely the number of page views and average time a user has spent on a page when predicting conversion. It is also important to include the different page types visited by users (Goldstein et al., 2017). Additionally, the sequential relationship in user behaviour is often not considered in traditional machine learning models. Finding relevant features that represent this sequential relationship is a difficult and time-consuming task. However, deep learning models are designed to handle this sequential dimension that traditional machine learning models have much difficulty with. This has been shown by how deep learning models can handle language, where context is very important in determining meaning or sentiment (LeCun, Bengio & Hinton, 2015).

Webb, Pazani and Billsus (2001) researched machine learning for modelling user behaviour over fifteen years ago. Back then, the authors identified four critical issues that limited practical applications of user modelling. These four issues were: the need for large datasets, need for labelled data, need for models to quickly adjust to changes in users and the need for computational complexity. Throughout the years many of these critical issues have been resolved. For example, many companies collect a lot of user data nowadays that is already labelled upon collection. Additionally, machine learning models can quickly be retrained to consider changes in a user base. Computational complexity was the fourth critical need that made practical user modelling impossible for a long time. However, because of the use of Graphical

Processing Units (GPU) the training of deep learning models has become practical (Coates et al., 2013). Services like Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform have made it easier to work with deep learning without big upfront investments. To further understand the dataset that this research will use, the company where the dataset has been collected will now be presented.

1.1 StudyPortals

This master thesis has been written in conjunction with an internship at StudyPortals in Eindhoven, the Netherlands. The internship had a duration of five months and took place from the 5th of March 2017 till the 5th of August 2017. StudyPortals is a global study choice platform that aims at making study choice transparent on a global level. The company was founded because of student problems. For example, the difficulty in finding the right international experiences. Unclear and different websites per university often make this process much longer and tougher than it should be. StudyPortals provides students with one platform where they can compare and save studies they find interesting. Therefore, StudyPortals employs different portals such as MastersPortal, BachelorsPortal and PhDportal. Because of this, StudyPortals has a big amount of user data. There is click data from everyone who has visited the website. On its portals, StudyPortals list a combined 140.000 courses from 2450 universities located in 68 countries. The number of registered users is around 2 million and since 2013 around 53 million page views on studies are listed in the database. As StudyPortals is a study choice platform it does not sell anything to its main visitors, which are students. However, the site does offer students the opportunity to click through from a study page to the website of the university. Students who take this action are 'converted' and represent students who go from the aware/informed stage in the conversion funnel to the interested stage. Figure 1 presents an example of a conversion funnel for a university listed on StudyPortals. In this case we are interested in predicting the 'red' students from our overall visitors. These users are the converted users. The conversion path can be different for each user, but often starts with them landing on the homepage of one of StudyPortals' portals. From there they can search for studies, or show studies them by discipline or country. Students then select a study and visit the page for that study. From that study page they can click-through to the university website and thus convert.

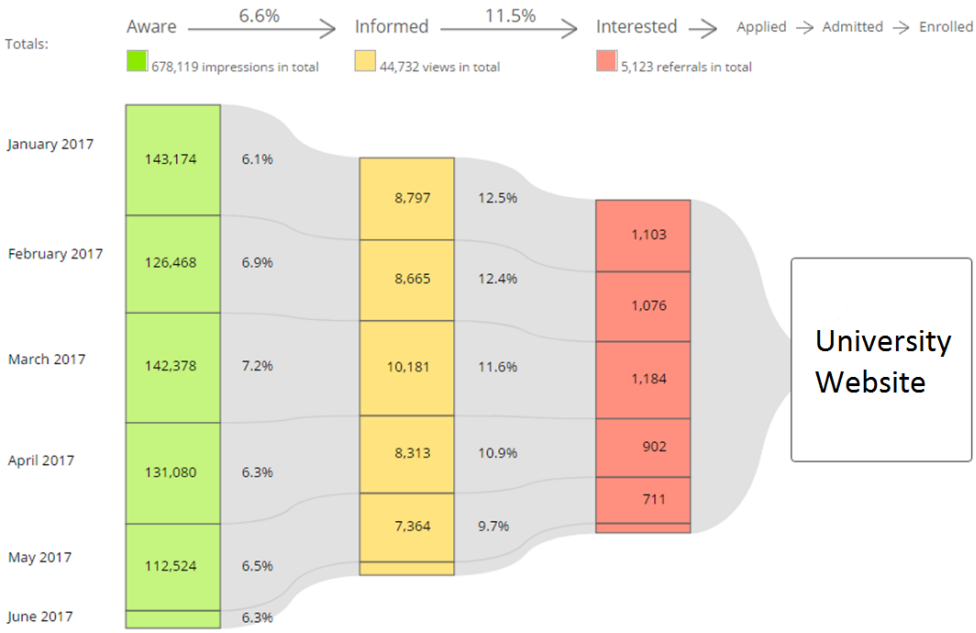

Figure 1. Example of a conversion funnel for a campaign at StudyPortals.

StudyPortals operates in the higher education market, which is currently going through a crucial period and is one of the main drivers behind the quick growth of StudyPortals. In 2013 a critical report was published on the state of higher education. According to the authors of this report “deep, radical and urgent transformation is required” (Barber, Donelly, Rizvi & Summers, 2013). Current models of education are said to be broken and universities are innovating too slowly to keep up with the rapid changes in the higher education market. These upcoming changes in higher education get compared to an avalanche. This comparison refers to how everything looks perfectly fine on the outside. However, on the inside all these changes are happening. Huge changes are coming although no one can say when they will come.

One of the main drivers behind these changes is globalization. Students are increasingly looking for education outside of their home country to find the best education. This development is caused by education-related factors like the increasing cost of higher education and how the economic value of a degree is dropping (Barber et al., 2013). Additionally push-factors like limited places for higher education in certain countries play a role as well. External factors like Brexit, the Turkish coup d'état and the Greek government-debt crisis also play an important role in students seeking for education across borders. StudyPortals has seen the effects of these events in their daily page visits. In general, international student mobility has been experiencing a continuing growth and has tripled between 1990 and 2014. An especially steep increase can be seen between 2000 and 2014, where international student mobility grew from 2.1 to 5 million (ICEF, 2015). Therefore, it becomes increasingly important for universities to reach international students and to track where each student is in the conversion funnel. This includes the ability to predict what users are most likely to convert and which users might not be likely to convert. Specific groups of users can then be reached through various interventions. Deep learning models seem to be a promising tool for predicting this user conversion.

1.2 Research Questions

The practical contributions of this study are found in various areas. Contributions will be brought to research that focuses on predicting and understanding conversion. This study will also generate insights into a more general understanding of how and when deep learning is an appropriate tool to use in marketing. Additionally, understanding the relationship between user behavior and conversion is important to StudyPortals. This will also be the first application of deep learning within the company. Other benefits include the enhancing of StudyPortals’ analytical capabilities and understanding of the predictive power of their platform. This study contributes to society as well. Deep learning has gathered a lot of hype surrounding itself. However, it is sometimes unclear how these models can be applied in practice. From a theoretical standpoint, there are multiple literary deep learning studies that suggest the usefulness of deep learning in marketing. However, there are few empirical studies that bring these opportunities into practice. Instead, studies in the research area of marketing often focus on statistical analyses. With the rise of deep learning models and the continuous growth of overall machine learning models it also becomes increasingly important to have a methodology to compare these various models. This thesis explores how to handle different input types and pre-processing, different metrics to use and takes into account architectural decisions when implementing deep learning. Through the combination of various metrics that describe different parts of the deep learning models like quality and usability, a type of quality testing model originates. The goal of this thesis is not to design such a model. Although the conclusion will reflect on the chosen metrics and discuss if they could be used to further built such a quality model. To better understand the usefulness of deep learning models they will also be compared with traditional machine learning models. Because there are few machine learning studies in marketing, this study will explore research in the field of Educational Data Mining (EDM). This field is close to our

data since both work with a dataset that consists of student data. The insights on machine learning algorithms in the EDM field will be used to select relevant machine learning models that will be used for comparison in this study.

This research will explore the value of deep learning when predicting conversion. In order to get a complete understanding of their strengths and weaknesses, multiple deep learning models will be compared with traditional machine learning models. The dataset these models will be tested on consists of clickstream data of students looking for an education abroad. First, the performance of traditional machine learning models will be compared on this dataset by using multiple metrics. Second, deep learning models will also be compared on this dataset. Third, the best performing models will be tuned through a process called hyperparameter optimization to decide which type of models perform best on this prediction task. Performance here does not merely reflect on the predictive quality of the model but also on its practical usability.

Therefore, the **research goal** has been defined as: *To capture the value of deep learning models for predicting customer conversion.*

Based on the introduction and the research goal, the following **research problem** has been formulated: *What is the value of deep learning models for predicting customer conversion?*

The following **sub-questions** have been formulated based on the research problem:

- *What variables has previous literature identified as being significantly related to conversion?*

In order to enter a dataset into a machine learning model, feature extraction has to take place first. Feature extraction turns the raw clickstream data into features that can be entered into the model. Variables that have already been identified as being related to conversion will play an important role in deciding what features to use. This question will be answered trough the literature review.

- *What are relevant metrics in the comparison of traditional machine learning and deep learning models?*

An important part in the comparison of the various models is deciding what metrics they are compared on. Instead of comparing them on a single metric, like accuracy, this research provides a more holistic comparison by including different metrics. The literature review will provide an overview of metrics used by previous research on which the metrics used in this study will be based.

- *What pre-processing steps should be taken in order to compare traditional machine learning and deep learning models?*

The different models used in this study require different sets of pre-processing tasks to optimally make their predictions. In order to properly compare each model it is important that the optimal combination of pre-processing tasks is used for each model. Previous literature will suggest what pre-processing tasks should work best for each model. Additionally, the best combination of pre-processing tasks will be tested in the methodology chapter through small experiments.

- *Do deep learning models perform better in predicting customer conversion than traditional machine learning models?*

To answer the research problem it is crucial to know if deep learning models perform better than the traditional machine learning models. This question will combine results of the comparison on all included metrics and will be answered through the empirical study.

The remainder of this research is organized as follows. Section two describes the process of the literature search, the literature review and ends with the research model. Section three presents the methodology used to test this research model. Section four presents the results from the analyses and comparison of traditional machine learning models and deep learning models. Finally, section five presents the conclusions and limitations of the study and will provide recommendations for future research.

2. Literature Review

This chapter will present the literature review. First, the literature search will be described in detail, followed by the literature review itself. The literature review starts with previous research in the area of predicting conversion. Next, studies where machine learning models and deep learning models are compared are discussed. Following, validation and performance metrics are described. Finally, the research model is presented.

2.1 Literature Search

A systematic literature review has been performed according to the methods of Wolfswinkel, Furtmueller and Wilderom (2013) and Webster and Watson (2002). These methods focus on transparency of the literature review and allow for reproducibility. A computer search has been conducted during May and June of 2017 on the international research databases Scopus and Web of Science. Only journal articles and conference papers were considered for inclusion in the literature review. The final selection of papers has been composed through comparing abstracts, removing duplicates, number of citations, forward and backward citations and finally reading the full texts. This process is described in more detail below.

Figure 2 represents the amount of conference papers and journal articles on 'machine learning' per year since 2000. Figure 3 represents the same for papers on 'deep learning'. Publications on machine learning seem to follow a more organic growth although a steeper increase since 2012 can be seen. On the other hand, the number of publications on deep learning experiences a very steep growth since 2012-2013. This shows the popularity that deep learning has been experiencing recently. Not just in practice but also in academics.

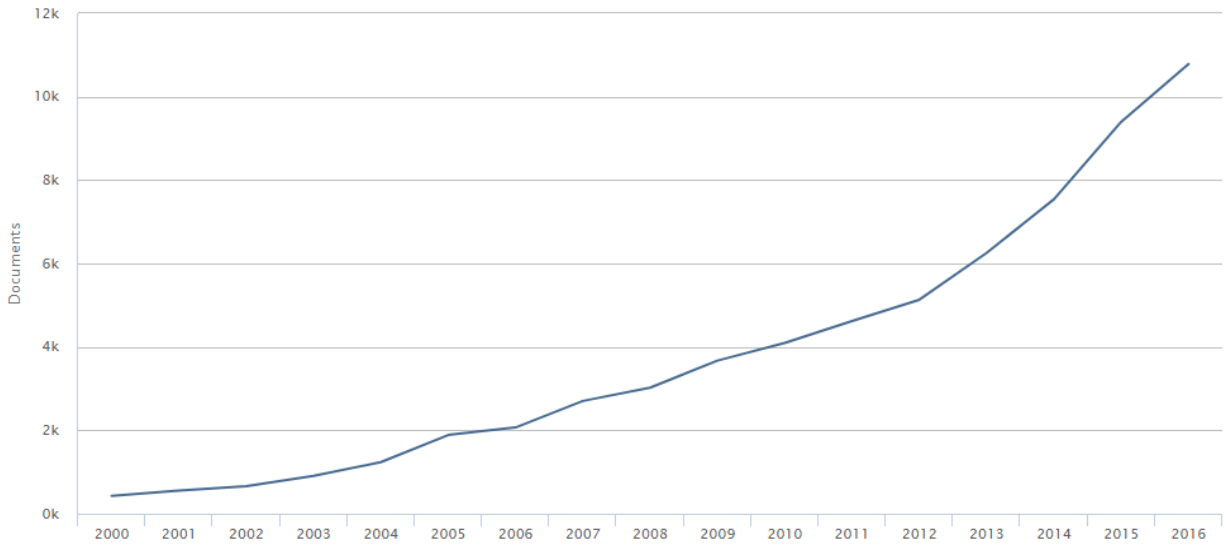

Figure 2. Number of journal articles and conference papers on 'machine learning' on Scopus per year.

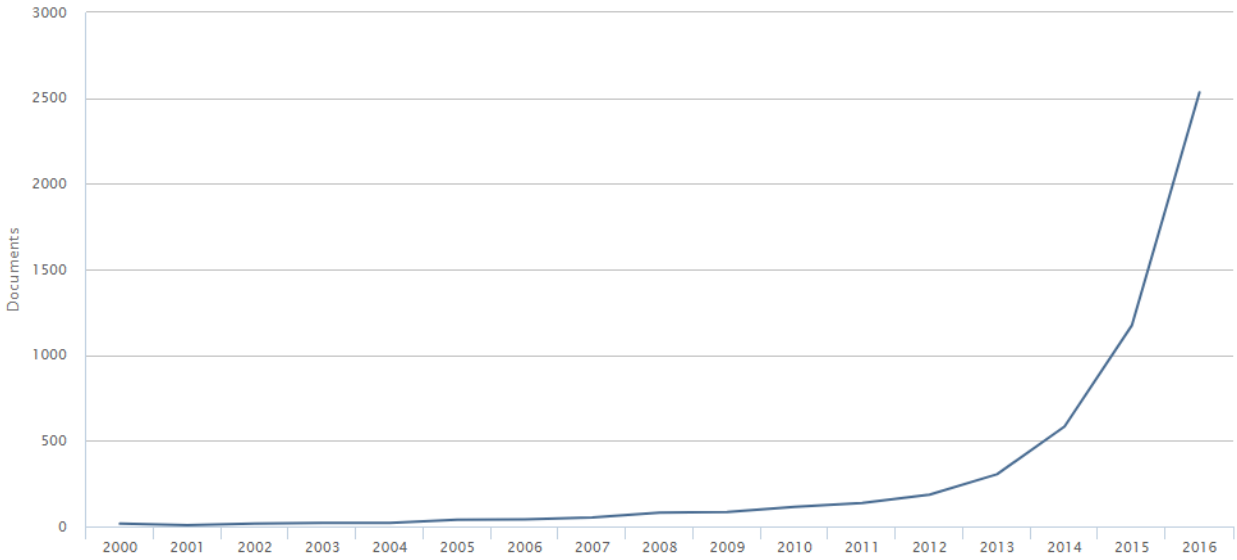

Figure 3. Number of journal articles and conference papers on 'deep learning' on Scopus per year.

Detailed explanations of why each search term has been used can be found in Appendix A. Articles for the literature review were selected as follows. First, a 'first' selection was made based on the title and abstract of the paper. During the search, there has already been controlled for duplicate papers between Scopus and Web of Science and between different search terms. Therefore, duplicates did not have to be removed afterwards. Next, inclusion was based on reading the full articles. After reading each paper a concept matrix has been updated, which can be found in Appendix B (Webster & Watson, 2002). Papers have been analyzed using the grounded theory approach, which has resulted in the current structure of the review (Wolfswinkel et al., 2013).

2.2 Modeling User Behavior for Conversion Prediction

Previous research has focused on the relationships between features like the number page views and session length on target variables like user conversion and likelihood to purchase (Bellman et al., 1999; Lin et al., 2010). However, recent research has shown that using more complex measures provides more accurate results when modeling user behavior (Goldstein et al., 2017).

Gündüz and Özsu (2003) were some of the earlier researchers to focus on the sequence of user behavior when evaluating clickstream data. Clickstream data refers to data that contains the path a visitor has taken throughout a website and reflects the choices made by the user. It can also contain the path of a visitor across multiple websites (Bucklin et al., 2002). The recommendation system designed by Gündüz and Özsu (2003) takes into account the sequence of visited pages and time spent on each page. Chan et al. (2014) use clickstream data and machine learning to predict whether and when to show a lead form to a user. Here, the authors include the type of pages that users are looking at when building the model. Goldstein et al. (2017) refer to this measure of different page types visited as search diversity. The authors focused on predicting conversion and find that as search diversity decreases the likelihood to purchase increases. These results are in line with previous research that showed that as consumers get further in the conversion funnel their searches become more focused. Additionally, this also proves the

importance of including different types of visited pages when modeling user behavior (Goldstein et al., 2017).

Lo, Frankowski and Leskovec (2016) researched user behaviors that led to purchasing on Pinterest in both the long and short-term. There are four type of actions a user can perform on Pinterest: search, zoom-in on a post, click through on a link and save a post. Results show that users with long-term purchasing intent tent to click through to more external content and save more posts. However, the closer a user gets to the purchase, the more their behavior switches from saves to searches. This indicates that the sequence of user behavior is indeed important when researching user behavior. On the other hand, Guo and Agichtein (2016) focus on the effects of mouse movement and scrolling in predicting whether the user has a research or purchase intent. As indicated by the results, scrolling interaction does matter when studying user behavior. Users that were predicted to have purchase intent had shown substantially higher amounts of scrolling than users with research intent.

These studies have shown that when modeling online user behavior, it is important to consider the number of page views (Bellman et al., 1999), session length (Lin et al., 2010), sequential aspect of user behavior (Gündüz & Özsu, 2003; Lo et al., 2016), different page types on the website (Goldstein et al., 2017; Chan et al., 2014) and scrolling behavior of users (Guo & Agichtein, 2016). Depending on the type of website studied these factors can all influence conversion.

2.3 Comparing Machine Learning Models

This part of the literature review will focus on the methodology behind machine learning algorithms and the comparison of them. The application of machine learning in Educational Data Mining research is studied because there are few examples of machine learning comparisons in a marketing context.

Universities and other educational institutions often have big collections of data on their students. To explore patterns and relationships in this data the research field of educational data mining started. Educational data mining (EDM) is a field of research that "is concerned with developing methods to explore the unique types of data in educational settings and, using these methods, to better understand students and the settings in which they learn" (Romero & Ventura, 2010, p. 601). Next to typical data mining techniques like clustering and classification, Romero and Ventura (2010) also include techniques like regression and visualization in their review on EDM. From a practical perspective EDM focuses on the discovery of knowledge through students' usage data. The discovery of most of these patterns has become impossible to do by humans due to the large amounts of data. EDM borrows from fields like statistics and machine learning. Research in EDM is often performed for specific educational institutions and therefore uses datasets that are often smaller than the typical datasets in machine learning. (Scheuer & McLaren, 2012).

Educational data mining consists of three key parts: pre-processing of the data, application of data mining techniques and post-processing (Romero & Ventura, 2007). Romero & Ventura (2010) their literature review shows that the main categories of research in EDM are student recommendations, predicting student performance and analyzing and visualizing data. Papamitsiou and Economides (2014) conducted a literature review with a focus on empirical studies and included 40 key papers in their review. The most popular method in the field was found to be classification followed by clustering and regression. Because of the nature of this research we are mainly interested in studies that apply machine learning for prediction and classification.

Dekker, Pechenizkiy and Vleeshouwers (2009) researched whether machine learning methods could be used to predict student dropout. The study was conducted for Electrical Engineering students at the University of Eindhoven and consists of 648 students. Various algorithms are compared including Decision Tree, Bayesian Networks, Logistic Regression and Random Forest. The one rule algorithm is used as a baseline to compare the other models against. Results show that the performance across most models is similar (79%-81% accuracy) and that the one rule algorithm (76%) and the Bayesian network (75%) perform worst. Only the decision tree based on the CART methodology provided a significant improvement over the baseline algorithm. In another study, Delen (2010) predicted student dropout using five years of freshmen student data. The prediction occurs at the end of the first semester so decision makers can potentially perform an intervention during the second semester. The dataset consists of 16066 students and 39 variables. The dataset is unbalanced and consists of 20% dropout students and 80% students that were retained. Artificial Neural Networks, Decision Trees, Support Vector Machines, Logistic Regression and various ensemble techniques are compared on predictive accuracy. When using the full (unbalanced) dataset the Support Vector Machine performs best with an accuracy of 87.23%. However, because the dataset contains much more retaining students than dropout students the model is not actually good at predicting dropout. Rather, the model is overestimating the amount of students in the retained category. Next, the authors create a balanced dataset where the Support Vector Machine still performs best with 81.18% accuracy. The authors also compared the ensemble methods Random Forest and Gradient Boosted Trees and found that Random Forest performed best.

Alper and Cataltepe (2012) compared machine learning algorithms on the task of predicting whether a student would pass or fail a fourth-year course based on previously obtained grades. The dataset consists of data on students from 2005 to 2011 for Computer Engineering students at Istanbul Technical University. Compared machine learning models include: Naïve Bayes, Neural Network (multilayer perceptron), SVM (rbf kernel) and Logistic Regression. The predictions are calculated for three different courses. For each course, a different model performs best. Naïve bayes, Logistic Regression and Bayesian Logistic Regression belong to the best performing models. On the other hand, Kabakchieva (2013) compared machine learning models while predicting student grades classified in 5 categories; bad, average, good, very good and excellent. Compared models are: Decision Tree, Naïve Bayes, Bayesian Net, k-Nearest Neighbors and Rule Learner. The accuracy in predicting different classes varies a lot. Additionally, all models have an overall accuracy below 70% as well as recall and precision scores below .70. The Bayes classifiers perform worst while the decision tree is the most reliable across all classes. The dataset is very unbalanced. For example, 4336 students are in the 'very good' class while 347 students are in the 'average' class. This is probably the cause of the bad performance of the machine learning models. Romero, Espejo, Zafra, Romero and Ventura (2013) also compare machine learning models while predicting student marks. The marks to predict are split-up in four categories: fail, pass, good and excellent. 21 different machine learning classifiers have been compared. Results show that pre-processing tasks like rebalancing of the data, discretization and processing of categorical variables all affect different models. However, the accuracy of most models is still not high, peaking below 66% accuracy.

Li, Wang & Wang (2017) use clickstream data to predict the final course grade of students of MOOC courses. The final grade is split in four categories. The authors address the issue of how 'traditional' machine learning models do not consider the sequence of user behavior. For example, if V stands for watching a video and Q stands for answering a question. Then it does not matter whether a user followed sequence V-V-V-Q-Q-Q or a user followed V-Q-V-Q-V-Q for most machine learning models. They will simply count this as: the user watched three videos and the user answered three questions.

The authors do not try deep learning algorithms but instead experiment with different n-gram features to use in the prediction. Results show that 3 or 4 n-gram features perform best when considering precision, recall and the f1-score.

2.4 Comparing Deep Learning Models

This section focuses on the use of deep learning in EDM research, how to compare deep learning models and the architectural decisions that have to be made when designing various models.

Guo, Zhang, Xu, Shi and Yang (2015) predicted student performance based on a combination of five types of data sources: background & demographic data, past study data, school assessment data, study data and personal data. Student performance is categorized in five classes. The deep neural network is first pre-trained using a sparse auto-encoder and is then treated as a supervised learning problem to finetune the parameters. The authors' deep neural network outperforms Naïve Bayes, SVM and multilayer perceptron algorithms. Piech et al. (2016) researched the use of Recurrent Neural Networks to model student learning. The main advantage of these type of networks is that they consider the sequential dimension and can model long-term dependencies. The input into the model is a list of student interactions and the output of the model consists of whether the student would answer exercises correctly or not. The authors use an embedding layer instead of one-hot encoding because one-hot encoding would result in very large and sparse vectors (Piech et al., 2016). The RNN with LSTM performs better than Bayesian Knowledge Tracing models. The two main advantages of this model are that they do not need expert feature engineering and they can operate on any input that can be vectorized. A downside is that they need large amounts of training data. Additionally, Tang et al. (2016) also researched the use of Recurrent Neural Networks with LSTM on sequential educational data. The authors tried the predict the next action of a student participating in a MOOC course. The best performing model was a Recurrent Neural Network with LSTM units. The model achieved an accuracy of 72.23%.

Tamhane et al. (2017) researched whether a sequence of visited product categories on a fashion e-commerce website could be used to predict the last viewed product category. To deal with the sequential nature of the data the authors use a Recurrent Neural Network with a Gated Recurrent Unit (GRU). The GRU has a similar structure as the LSTM units discussed earlier. The authors also use an embedding layer to map each product category into a vector. The RNN achieves better results than the baseline methods; majority voting and the product group graph method. Main conclusions are that the RNN (with GRU) performs better when the context changes within a session. Additionally, the RNN performs better as more data becomes available and when the behavioral user sequence becomes longer. Salehinejad and Rahnamayan (2016) used a RNN with LSTM to accurately predict customer behavior on a grocery shopping dataset.

In comparison to the traditional machine learning algorithms, deep learning algorithms do not have a standard architecture. Instead, the researcher has to define different steps in the design of a deep learning architecture. This includes the number of hidden layer to use, number of hidden nodes, dropout, optimizers etc. In order to compare the different deep learning models in this thesis they will all use the same 'architectural choices' wherever possible. Dropout is the process of randomly turning off weights in a neural network while training the network through backpropagation (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). Dropout is a very important practice in preventing neural networks from overfitting. Dropout is only applied when training the model, when predicting on the test

set or on other new data all neurons are 'on'. Figure 4 presents a visual representation of dropout. When comparing different deep learning models, it is important to keep activation functions and the number of hidden units similar across different models (Guo et al., 2015; Piech et al., 2016). Application of dropout is important to prevent complex architectures from overfitting soon in the training process.

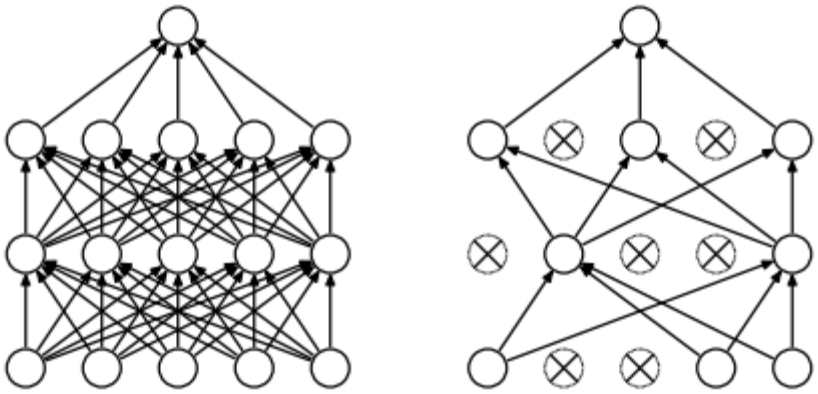

Figure 4. Example of dropout. Adapted from Srivastava et al. (2014, p. 1930).

In Natural Language Processing, it has become increasingly popular to transform words into a vector as preparation for a deep learning model. This approach was popularized by the 'word2vec' model and has been described in Mikolov, Chen, Corrado and Dean (2013) and Goldberg and Levy (2014). When using an embedding layer each word is changed to an index. In the embedding table this index can then be used to look-up the corresponding vector. Embedding layers do not result in huge vectors like one-hot encoding all these words would in a big language model. During the training process of the deep learning algorithms the weights of the embedding vector get updated just like the weights in the deep learning model (Piech et al., 2016). This also allows for the exploration of similar words after a language model has been trained. Lately, research has explored the use of embedding layers for different features than words or word combinations. Tang et al. (2016) used embeddings to represent actions taken by users in a MOOC. Additionally, Tamhane et al. (2017) used embeddings to represent different product categories on an e-commerce website.

2.5 Model Validation and Metrics

Next, it is important to consider the validation approach and metrics used when comparing different machine learning models.

Cross-validation is a common approach in the comparison of machine learning models (Alper & Cataltepe, 2012; Kabakchieva, 2013; Romero et al., 2013). In this process, the dataset gets split up in different folds for validation of the prediction model. The obtained metrics from the different folds are then averaged at the end. For example, when using 10-fold cross-validation the machine learning model gets trained on 90% of the dataset and validated on 10% of the dataset. This then happens 10 times for different folds of the dataset. This provides a more stable way of analyzing a machine learning model than using one training and test split. However, deep learning models are often much more expensive to train. Even when training deep learnings on a GPU they can take significantly longer to train than traditional machine learning models (Bengio, 2012). Therefore, cross-validation is much less used in

deep learning because the process is often too time consuming. In deep learning, the single train-test split is often used.

In terms of metrics to validate the model on, accuracy is the main metric used and occurs in almost every study where machine learning algorithms are compared. Additionally, recall, precision and the f1-score all provide additional insights next to accuracy (Kabakchieva, 2013; Li et al., 2017). Especially in unbalanced datasets, accuracy can give a distorted picture of the performance of the algorithm. The f1-score is a way to combine precision and recall into one metric. Dekker et al. (2009) chose a baseline algorithm and compared if other models significantly improved on the accuracy of the baseline model. Guo et al. (2015) and Tang et al. (2016) point out the importance of training time in the comparison of deep learning models. These models must be trained on a GPU. Additionally, training on a GPU can still take very long which can make it difficult to do a thorough grid search of hyperparameters. Table 1 presents an overview of the different metrics used in research that involved the comparison of machine learning models.

Table 1

Metrics used in papers where machine learning models are compared

Authors	Metrics used in study
Alper & Cataltepe (2012)	Accuracy
Dekker et al. (2009)	Accuracy, significant changes on FP, NP, TP, TN
Delen (2010)	Accuracy, per-class accuracy
Goldstein et al. (2016)	Precision, F1-score
Guo & Agichtein (2016)	Precision, recall, f1-score
Guo et al. (2015)	Accuracy, training time
Kabakchieva (2013)	Recall, precision
Li et al. (2017)	Precision, recall, f1-score
Piech et al. (2016)	AUC
Romero et al. (2013)	Accuracy
Tamhane et al. (2017)	Normalized Discounted Cumulative Gain, Precision, Recall
Tang et al. (2016)	Accuracy

2.6 Research Model

Previous research has shown that conversion is not only linked to measures like the number of page views and average time spent on a page. Additionally, it is important to also include what type of pages are being viewed, scrolling behavior and the sequential nature of this data (Gündüz & Özsu, 2003; Guo & Agichtein, 2016). It can also be important to include additional information like the country of the user (Chan et al., 2014). For the traditional machine learning models, the features to be included in the model must be pre-defined. These features are based on user behavior and will be used to predict whether a user converted or not. A schematic representation of this model can be found in figure 5. The user behavior for deep learning models will be represented as a sequence of user behavior. A schematic representation of that model is shown in figure 6. The models take in a set of input data, learn patterns from that data and use those patterns to predict whether a user will convert or not convert. Figures 5 and 6 also show which models will be compared. In chapter 3 the specific features will be able to be filled in to this model. Table 2 provides a more detailed overview of the compared models.

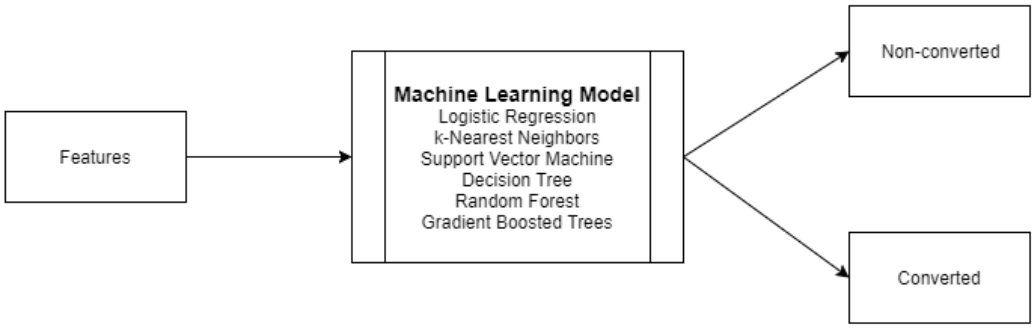

Figure 5. Schematic representation of the traditional machine learning approach.

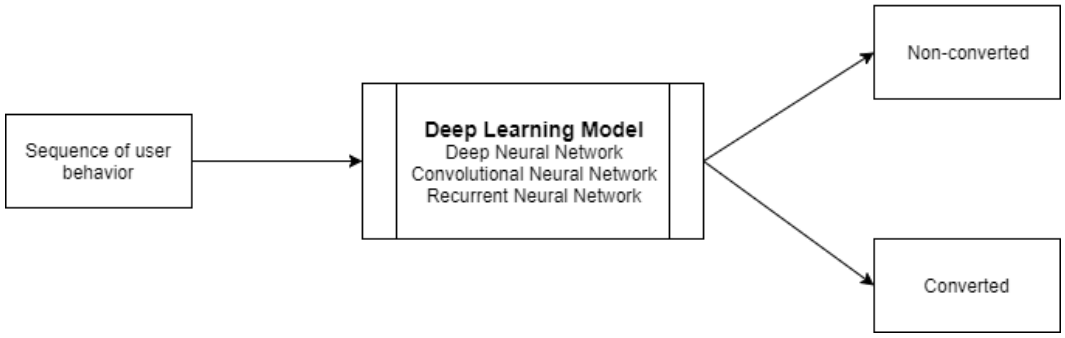

Figure 6. Schematic representation of the deep learning approach.

The goal of this study is to capture the value of deep learning for predicting customer conversion. In order to do that various deep learning models are compared with each other. Additionally, these deep learning models are also compared with traditional machine learning models. All these models are compared on the same metrics. Part of these metrics focuses on assessing the predictive quality of the model. These metrics are accuracy, precision, recall, f1-score and Logloss. The other part of these metrics focuses on the practical usability of the model. These metrics are the training time, prediction, the epoch the model achieved its best prediction at and the transparency of the model. The epoch metric is specific to deep learning models. Table 2 presents an overview of the table that this study will try to fill-in and is therefore also the research model of this study.

Table 2

Research Model

Model	Accuracy	Precision	Recall	F1-score	Logloss	Training Time	Prediction Time	Epoch	Trans-parency
Model A									
Model B									
Model C									
Model D									
Etc.									

3. Methodology

This study will compare various traditional machine learning models and deep learning models on predicting conversion. Both approaches require their own type of input data and specific pre-preprocessing tasks. Therefore, most methodology sections are split-up in two parts. This provides a clearer way to describe the required pre-processing steps and analysis required for both types of models. First, the modeling approach will be discussed which presents a short summary of the different models used in the study. Second, the dataset and its collection will be discussed. Third, the feature extraction process will be described. Fourth, the various pre-processing tasks will be discussed. Finally, the metrics and validation of the models will be discussed.

3.1 Modeling Approach

3.1.1 Traditional Machine Models

The traditional machine learning models require features to be extracted and their input to be prepared in a tabular format. The best way to understand this is by comparison to an Excel sheet. Every row is one sample of the data and every column is one feature. When working with a tabular dataset it does not matter whether a value is in row 1 or in row 100, column 1 or column 100. Therefore, the input into these models can include features like: the number of study pages visited, the total number of pixels scrolled etc. However, the model cannot consider the effects of what action was taken in what order. This section provides a quick overview of the different machine learning algorithms used in this study.

Logistic Regression

Logistic Regression is a regression model where the dependent variable is categorical. Standard logistic regression focuses on binary classification. The model tries to fit a regression curve to the data using the sigmoid function and predicts whether a data point belongs to category 0 or 1.

k-Nearest Neighbors

The k-Nearest Neighbors algorithm (k-NN) makes a prediction based on the similarity of nearby data points. The similarity of data points is calculated based on the distance between the features of each data point. Therefore, it is important that all features are on a similar scale. Otherwise, a single variable with high values can skew the prediction.

Support Vector Machine

The support Vector Machine (SVM) tries to find a multi-dimensional hyperplane that separates the data points in a way that maximizes the distance between this hyperplane and the closest data point on each side of the plane. If the input is a k-dimensional vector (where k stands for each feature), the algorithm operates in a k-dimensional space. The SVM supports various kernels which can be used so that the algorithm can find non-linear relationships in the k-dimensional space. In this thesis, the linear kernel and the radian basis function (rbf) kernel will be used.

Decision Tree

A decision tree is a common structure used in decision making processes. At each point in the decision tree a split is made based on a feature in the dataset. A data point ‘travels’ along the tree and each decision node determines where the data point will end.

Random Forest
A random Forest (RF) is an ensemble method of decision trees that constructs many decision trees based on random subsets of features and random parts of the dataset. One of the main advantages of a random forest is that the random selection of features on each tree balances the tendency of decision trees to overfit on the training set. Overfitting occurs when a machine learning model is very good at finding patterns in the provided training set but does not generalize well to new data.

Gradient Boosted Trees
Gradient Boosted Trees (GBT) is another ensemble method of decision trees. The core idea behind gradient boosting is to combine many 'weak learners', decision trees that perform not much better than random guesses. Each new tree is added on top of existing trees and a loss function is minimized through gradient descent. This thesis will use the 'XGBoost' implementation of Gradient Boosted Trees which uses more regularization to prevent overfitting.

3.2.1 Deep Learning Models
Deep learning models have been achieving breakthroughs with unstructured data. Examples of these data types are images, speech and text. These data types all include a sequential element that traditional machine learning models cannot easily capture. For example, when working with image recognition it is very important to know in what part of the image a certain group of pixels occurs. If you try to recognize a face it is not enough to know that there are two eyes, a nose and a mouth in the image. Additionally, you also need to know where in the image these things are in relation to one another. Another example is found in natural language processing. When working with sentences the order in which words appear is very important in determining the meaning and sentiment of the sentence. Next to language, behavior can also be organized as a sequence.

Deep Neural Network
The Deep Neural Network (DNN) is an artificial neural network with multiple hidden (deep) layers. The combination of multiple deep layers allows the network to learn complex and non-linear functions. Basically, the network consists of matrix multiplications along each layer of the network. The network is trained through backpropagation with gradient descent.

Convolutional Neural Network
The Convolutional Neural Network (CNN) is a popular choice in image recognition. The network operates by sliding multiple filters over the data. The contents of these filters are learned by the network itself. In the case of images, the inputs are multi-dimensional. However, in this case the input is a one-dimensional sequence. Therefore, the filters are one-dimensional as well.

Recurrent Neural Network
The Recurrent Neural Network (RNN) is a neural network that operates in a directed cycle. Therefore, these types of networks are useful when using sequences as they can learn long-term dependencies. DNNs assume that all inputs are independent of each other. However, the RNN takes into account what inputs came before the current input and saves this in its 'memory'. The Long Term Short Term (LSTM) application of RNNs has a different way of calculating the memory of the model. LSTM cells calculate what input the model should keep and what it can forget.

The DNN, CNN and RNN architectures have been chosen because they represent the main architectures used in deep learning (LeCun et al., 2015). Different variations of each architecture will be used related to the number of hidden layers and type of units used in the recurrent neural network. The architectural choices in deep learning are, for a big part, still based on intuition. By exploring different variations of each architecture the effects of different architectures can be seen.

3.2 Data Collection

All data for this master thesis has been collected from StudyPortals' database. StudyPortals stores their data in Amazon Redshift. Redshift is a cloud data warehouse system hosted by Amazon and is built on PostgreSQL. StudyPortals tracks and stores user behavior data in table called 'events'. This events table contains clickstream data of the users of the StudyPortals website The two main types of events that are stored in this table are 'page view' and 'page ping' events. Page view events show which pages the user viewed and can be used to construct the path a user took on the StudyPortals website. Page ping events are recorded per interval of a couple seconds and show the position of the website on the user's screen. From this information the scrolling behavior of the user can be calculated. The events table also records the conversion events that this study will try to predict. The full events table has a very big amount of data. Therefore, the data analysis for this thesis will be executed on a sample of this dataset. Training some of the machine learning models on the full dataset would be too heavy to run locally and would also take too long. Therefore, the comparison of models is done on one month of data ranging from 25-04-2017 till 25-05-2017. This sample contains a total of 56.349.410 events and includes over 36.000 converting users. From this sample, the user behavior for both converting users and non-converting users can be extracted. To create the dataset of converted users the first step is to get the first conversion event for each user in the user's first session (Lo et al., 2016). Next, the behavior for each user up until the conversion event is collected based on the timestamp of the conversion event. The process to create the dataset of non-converting users is similar. Although there, only users without conversion events are included since users without these are non-converting users. The behavioral dataset is then created by combining the dataset that has information of the converted users with information of users that did not convert.

For the training of all machine learning models a balanced dataset is used. This means that the same amount of converting users as non-converting users will be used to train the models. There are much more people in the overall dataset that did not convert than users that did convert. We take the number of converting users and then randomly select the same number of non-converting users. Using the full amount of non-converting users would greatly skew the dataset as seen in Delen (2010). For example, imagine a dataset consisting of 95% non-converting users and 5% converting users. A machine learning model can then simply predict that a user is non-converting 95% of the time and achieve 95% accuracy without learning anything from the data. Additionally, only users that have at least three registered page views are included in the dataset that is used for training the machine learning models. A quick first model built as a test model scored 94% accuracy. However, this model was greatly skewed by non-converting users with 0 registered page views. These users are probably logged into the table by another one of StudyPortals' databases. Because the referral button is shown on study pages, a user needs to have page views to see a referral button and click on one. See appendix C for more information on this decision.

Retrieving data from the Amazon Redshift database will be done using SQL. Data manipulation, analyses and building the machine learning models will be done in Python. The traditional machine learning

models will be trained locally on a CPU. The deep learning models will be trained on an external GPU server hosted by Amazon. These GPUs are optimized for deep learning models and allow for great parallelization of matrix operations.

Python libraries that are the standard for data science have been used in this research. Here is an overview of the main python libraries used and the task that they were used for:

- Numpy (linear algebra)
- Pandas (data analysis)
- Matplotlib & Seaborn (data visualization)
- Scikit-learn (machine learning)
- XGBoost (gradient boosted trees)
- Keras & Theano (deep learning)
- SQLAlchemy (communication with StudyPortals' Redshift database)

3.3 Feature Extraction

3.3.1 Traditional Machine Learning Models

For the traditional machine learning models, the feature extraction process is crucial for the performance of the model. This section presents how all features have been extracted from the raw clickstream data. Table 3 presents an overview of all the extracted features. The features are presented in categories that show the type of data each feature belongs to. Most features are selected based on results from previous studies discussed in the literature review. Other features are specific to the StudyPortals website and are therefore extracted. It should be noted that the main goal here is to extract as many meaningful features that could influence conversion. This study does not test hypotheses for the relationships between each feature and the target variable and whether these are significant or not. The main focus is on the comparison of the different models.

Total interactions

The first feature simply contains information on the overall interactions recorded per user.

Online tests

StudyPortals offers two online tests that users can fill in to assist them with their study choice. These tests are a country test and a personality test. The tests serve as a way to give the user better suggestions for studies that might fit them. Doing one of these tests could influence the conversion of a user. It is expected that a user that got tailored suggestions is more likely to convert. These variables are measured as binary variables: whether a user did the test (1) or not (0).

Page pings

Next, features related to the page ping vents are extracted. Page ping events are recorded in regular intervals when a user is on the website. Page pings contain information on the starting point and end point of the website on the user's screen. From that information, the scrolling behavior of the user can be inferred. The StudyPortals website is designed for vertical scrolling only. If a user is not scrolling the page ping will shows that the user has the same starting point and end point in that interval. This could, for example, mean that the user was focused on reading the page. The number of page ping events and the sum of vertical scrolling in pixels are included as features in the model.

Page views

The next type of events to include in the model are page view events. The total number of page view events is included as a feature. Next, the average time spent on each page has been calculated by dividing the session length by the total number of page views per user. The average time per event contains information on whether the user has stayed longer on each page or whether the user is quickly clicking through the website. The URL of the viewed page can be used the extract the type of page the user viewed. There are many different pages a user can visit and some of them are quite rare. Including all these unique page types into the model would cause the model to be flooded with redundant features (Chan et al., 2014). In order to make the model more transparent and reduce unnecessary complexity of the model only the most viewed page types are included. The most viewed pages in a recent daily sample of the dataset were explored. Table 4 presents the top ten page types that are included for feature selection. It was decided to combine 'study-options' and 'study-options-c', as well as 'scholarship' and 'scholarships'. In both cases these pages represent the same page type but are logged differently depending on the portal the user is on. The sum of page views on the eleventh most visited page type was a lot lower than on the 10^{th} most viewed page type. (from 371.169 to 89.023). Therefore, the top ten pages are included as features for the model. The page type can be extracted from the URL by splitting the URL based on slashes and selecting the part between the first and second slash. For example, the page URL is saved as 'www.mastersportal.eu/studies/29143/sciences-du-medicament-qualite-des-medicaments-et-des-aliments.html'. From this example 'studies' is the page type. The number of page views per page type are also included as features. These are the page types shown in table 4.

Meta information

Finally, the meta information from each user is included. These features are included as categorical features and include information on the user's: country of origin, browser, operating system, type of device and the portal the user entered on. StudyPortals has various portals (BachelorsPortal, MastersPortal etc.) and the portal the user visited could influence conversion.

Target

The variable we are trying to predict is referred to as the 'target'. This variable measures whether a user converted or not. 'Target' is a binary variable, 1 if the user converted and 0 if the user did not convert. This feature is calculated by checking whether a conversion event has been recorded for the user.

Table 3

Features included in the dataset used for the Traditional Machine Learning models

Feature	Description	Example values
Total interactions		
No_of_interactions	Total number of interactions.	integer
Online tests		
No_of_ctests	Whether the user did a country test.	0, 1
No_of_ptests	Whether the user did a personality test.	0, 1
Page pings		
Page_pings	Number of page ping events	Integer
Scrolling	Total amount of vertical scrolling in pixels.	Integer
Page views		
Page_views	Number of page view events.	Integer
Avg_page_time	Average time per interaction in seconds.	Integer
No_studies	Number of study pages viewed.	Integer
No_searches	Number of search pages viewed.	Integer
No_homepages	Number of homepages viewed.	Integer
No_study_options	Number of study option pages viewed.	Integer
No_universities	Number of university pages viewed.	Integer
No_disciplines	Number of discipline pages viewed.	Integer
No_articles	Number of article pages viewed.	Integer
No_countries	Number of country pages viewed.	Integer
No_account	Number of personal account pages viewed.	Integer
No_scholarships	Number of scholarship pages viewed.	integer
Meta information		
Page_url_host	The portal that the user entered on.	www.mastersportal.eu
Geo_country	Country the user visited from.	GB, US
Br_family	Type of browser used.	Firefox
Os_family	Type of operating system used.	Linux, Chrome OS
Dvce_type	Type of device used.	Computer, Mobile
Target		
Target	Whether a user converted or not.	0, 1

Table 4

Top ten most visited page types in the sample

Page type	Page views
Studies	2.323.800
Search	1.439.348
Homepage (empty string)	1.061.880
Study-options + study-options-c	1.072.820
Articles	888.503
Universities	731.369
Disciplines	586.254
Countries	529.042
Account	466.843
Scholarship + scholarships	371.169

3.3.2 Deep Learning Models

For deep learning models the feature extraction part is much less crucial to the performance of the model. Deep learning models essentially learn the features themselves. Therefore, these models can use much 'rawer' data than traditional machine learning models. However, the sequence of clickstream data as stored in the events table cannot be directly passed into them either. In order to create the behavioral sequence per user, decisions have to be made as to what constitutes a behavior that should be included in that sequence. Categorical variables like the browser of the user and the country of the user are not included here. The focus is purely on the behavior of the user as this can be represented as a sequence. As mentioned in the previous section the events table at StudyPortals mainly tracks page view events and page ping events. Therefore, behaviors are extracted from these two event types. The different type of behaviors will be represented by an index to be used by the embedding layer in the model (see section 3.4.2). Therefore, decisions need to be made on what constitutes a behavior and which behaviors should be included.

Page view events

A page view event is characterized by two main variables: the portal the user was on and the type of page the user viewed. Portals refer to different websites of StudyPortals like BachelorsPortal and MastersPortal. The type of page refers to different pages users can visit on these portals like studies, articles and disciplines. By combining portal type and page type the different page view behaviors will be created. It is expected that different pages will be popular on different portals. Also, portals can be organized differently and whether a user will convert could be influenced by the portal they are on. Similar to the traditional machine learning models we do not want to include every possible combination of portal and page type. Therefore, we look at the most viewed portals and page types in our dataset and use this to decide which portals and page types are viewed enough to include. For the portal type, there are 13 main portal types until page views per portal drop quickly. Portal types not part of these 13 are combined into a 14th type called 'other'. Similarly, there are 25 main page types, the other ones are labeled as 'other'. For the traditional machine learning models features were included for 10 different page types. Because the deep learning models treat these as behaviors in a sequence instead of individual features more levels of page types can be included for the deep learning models. As an example these page view events have now been turned into behaviors like 'mastersportalstudies' which refers to a page view on a study page on MastersPortal.

Page ping events

Page ping events include information on vertical scrolling in pixels. Instead of calculating the sum of pixels scrolled, we now include each scrolling behavior in the behavioral sequence. To code this scrolling into behavior they have to be put into categories of pixels scrolled. Figure 7 presents the histogram on the amount of scrolling per page ping event. There is a small number of outliers above 5000 pixels that are excluded from this histogram for visualization purposes. One can see that the number of page ping events with over 4000 pixels scrolled is very small. Therefore, it has been decided to create scrolling categories up until 4000 pixels. The smallest category is 0 scrolling and everything above 4000 is categorized as 4000+. Everything in between is split into categories of 100 scrolling. The page ping events used to calculate the scrolling behavior are recorded in regular intervals. Zero pixels scrolled means that the user has not scrolled in that interval. This can, for example, happen when a user is reading text on the page. As an example the categories of scrolling are saved as '0' and '200-300'.

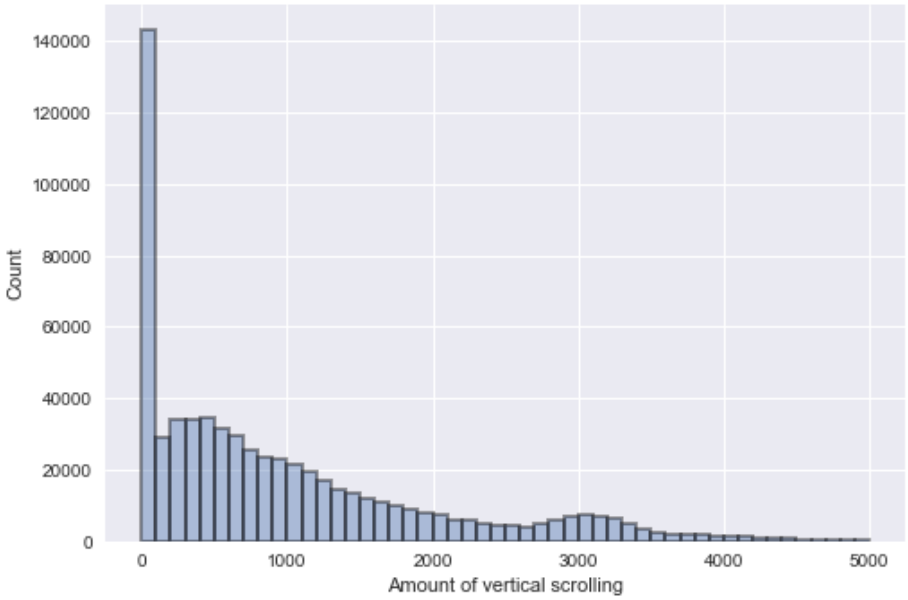

Figure 7. Histogram on vertical scrolling in pixels per page ping event.

3.4 Data Pre-processing

3.4.1 Traditional Machine Learning Models

To properly compare different machine learning models, they should be compared on their base parameters (Kotsiantis, Patriarcheas & Xenos, 2010; Delen, 2010). However, most literature is unclear about the input data that goes into the model. Different machine learnings need their input to be prepared in a different way. Therefore, it is important to explore how the features should be pre-processed before entering the model. Not all machine learning models can deal with categorical variables in a similar way. Additionally, certain machine learning models make decisions based on the distance between features. For these models, it is important that the inputs are on the same scale. Otherwise one variable with higher values can bias the whole prediction.

Categorical Variables

There are two main ways to encode categorical variables in machine learning: label encoding and one-hot-encoding. Label encoding turns categorical variables into numbers. Tree-based machine learning models can handle these categories because of how they split the data at each node. However, linear models cannot deal with this encoding because it implies an order. For these linear models, another type of encoding is often used called one-hot encoding. One-hot encoding creates a new feature for each level of the categorical feature. For the true value that feature will be 1 and for all other features it will be 0. When a categorical variable has many levels this will create many of these 0 values in the dataset. Table 5 provides an example of both types of encoding using the device type of users.

Table 5

Example of Label Encoding and One-Hot Encoding

		One-Hot Encoding		
	Label Encoding	Is_computer	Is_tablet	Is_mobile
Computer	0	1	0	0
Tablet	1	0	1	0
Mobile	2	0	0	1

Normalization

Additionally, some machine learning models operate on the distance between features. For example, the k-NN algorithm searches to find the closest neighbors to a data point based on the distances between its features. Therefore, it is very important that these features are on a similar scale. Normalization takes care of this issue by scaling all numerical features to be on a scale from zero to one.

Literature suggests how the categorical variables should be encoded and whether normalization should be applied based on the type of machine learning model. For example, linear models like logistic regression and SVMs cannot handle label encoded variables because it interprets these as having an order. On the other hand, tree-based models can handle this encoding because these models are not linear. To test the right combination of pre-processing a small experiment was conducted. A balanced sample of the dataset was taken consisting of 10.000 converting users and 10.000 non-converting users. The different types of traditional machine learning models have been trained on this sample with the input features pre-processed in different ways. Table 6 presents the results of this test and shows the accuracy score per algorithm and pre-processing combination. This table confirms the expectations that the k-NN algorithm needs both normalization and one-hot encoding. It also shows that the decision tree performs best when it only uses label encoding. In the results chapter, each of these models will be provided with the input that it shows the highest accuracy with in table 5. The random forest and gradient boosted trees models are ensembles of decision trees. These models will use the same input as the single decision tree.

Table 6

Accuracy scores (%) of different pre-processing tasks per machine learning model

Model	LE	LE & Normalization	OHE	OHE & Normalization
Logistic Regression	74	69.32	**75.85**	71.88
k-NN	55.85	66.68	59.5	**68.57**
SVM (linear)	63.43	70.87	70.87	**72.37**
SVM (rbf)	53.63	59.2	59.85	**66.93**
Decision Tree	**77.23**	75.87	76.77	77

Note. LE = Label Encoding. OHE = One-hot Encoding. The highest accuracy score per model is in bold.

3.4.2 Deep Learning Models

Now that all unique behaviors have been defined in section 3.3.2 it is possible to assign an index to each unique behavior. For example, a page view on a study page on MastersPortal gets index 1 and scrolling between 200 and 300 pixels get index 2 etc. Using these indexes, a numerical sequence representing the behavior of the user can be created.

Sequence length

Next, a decision should be made on the maximum length of the behavioral sequence. Some users have a short sequence of behaviors while other users show much more behaviors. The deep learning models expect all input to consist of the same length. Therefore, the sequence length must be decided. Figure 8 provides a histogram on the sequence length per user, where the x-axis has been cut at 250 for visualization purposes. From this histogram it has been decided to use a sequence of 125 behaviors per user, because after this number the user count drops and becomes very close to zero. This means that for users with more than 125 behaviors only the first 125 will be used to train the model. Users with less than 125 behaviors will have their behavioral sequence padded with a ‘zero’ behavior that the deep learning models can interpret as a masked value.

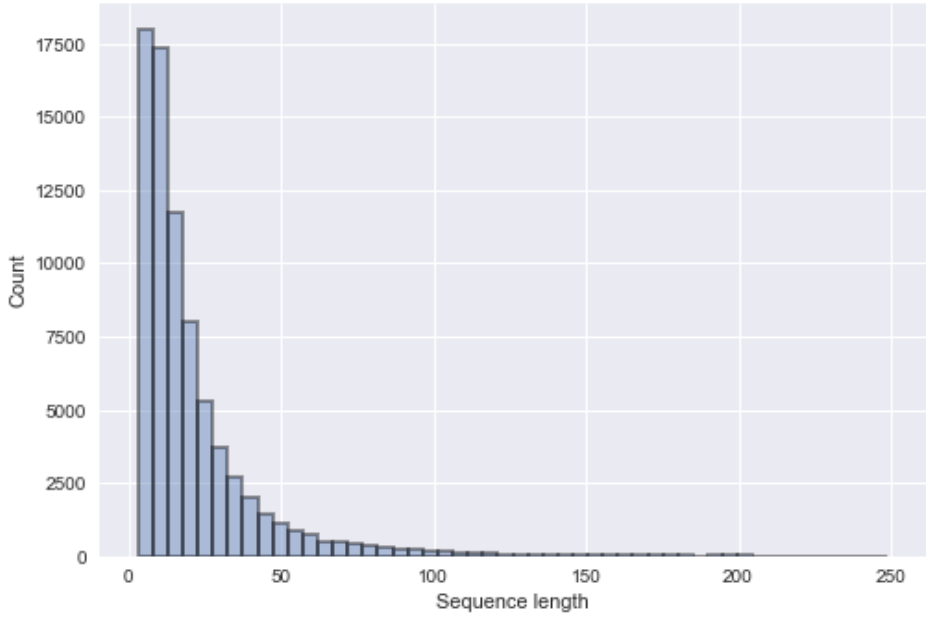

Figure 8. Histogram on sequence length per user.

The sequence of user behavior has been created using the indexes for each unique behavior. Therefore, an example sequence looks like this: 4, 142, 3, 3, 7, 47, 59, 101, 45 etc.

All deep learning models will start with an embedding layer (Tang et al., 2016; Tamhane et al., 2017). This layer assigns several latent factors to each behavior. These latent factors are then used to train the model. Basically, it turns positive integers into dense vectors of fixed size. The deep learning model can be trained using these latent factors (Gal & Ghahramani, 2016). Table 7 provides an example of a small embedding matrix with five unique behaviors and 9 latent factors per behavior.

Table 7

Example of a small embedding matrix

		Latent Factors								
Index	Represented behavior	LF1	LF2	LF3	LF4	LF5	LF6	LF7	LF8	LF9
1	Mastersportalstudies	.32	.02	.48	.21	.56	.15	.21	.27	.23
2	Bachelorsportalsearch	.65	.23	.41	.57	.03	.92	.88	.36	.24
3	Phdportalcountries	.45	.87	.89	.45	.12	.01	.51	.14	.74
4	0-100	.65	.21	.25	.45	.78	.82	.36	.34	.23
5	1500-1600	.04	.19	.63	.27	.85	.15	.56	.15	.14

Additionally, all deep learning models will include dropout (regularization). Each deep learning model will be run for ten epochs and metrics will be reported for the best epoch. An epoch is one full pass through the dataset. All models will use binary cross entropy for the loss function, the Adam optimizer, a batch size of 64 and will have 32 latent factors in the embedding layer.

3.5 Model Validation

The dataset will be split-up into a training set and a test set. Machine learning models are often compared based on cross-validation. However, for deep learning models this is often not possible due to computational restrictions. Therefore, this research will use the same train-test split for all models. The training set is used to train the model on. This where the model learns and finds patterns in the data. However, there is the risk that the model overfits. This happens when the model learns patterns from the data that are too specific. The model then does not generalize well when making predictions on new data, which is the main reason machine learning models are used. In order to the test the generalizability of the model a test set is used. The test set includes data that the model has not seen before. The training set will contain 70% of the data and the test will contain 30% of the data. Splitting the dataset in this way allows for a good general comparison of the different algorithms. Next, the best performing models will be tuned through a process called hyperparameter optimization. The hyperparameter optimization will happen through a grid search. A grid search is a way of optimizing hyperparameters by trying different values of these parameters. Machine learning models have a set of parameters that are learned from the training data. Additionally, there are other parameters that cannot be learned from the training process. These parameters are called hyperparameters and vary for each machine learning model. This process allows us to see how much performance gain can be achieved by not simply using the base values of the hyperparameters of these models. The differences in performance will probably be small based on changing the hyperparameters. Therefore, cross-validation will be used when tuning the parameters of the best performing models. Cross-validation splits the dataset in different training and test sets and calculates the performance for each fold. Then, the performance scores are averaged at the end. By using k-fold cross-validation we know that the results are not influenced by the way the dataset is split.

Evaluation Metrics

Before introducing the evaluation metrics the models will be compared on, it is first important to introduce the confusion matrix in table 8. The confusion matrix shows what the model predicted as the target variable and what the actual value of the target variable is. Based on the confusion matrix the evaluation metrics can be calculated. The main evaluation metric will be accuracy, which is simply the percentage of correctly classified users. Because all algorithms will use a balanced dataset, accuracy is a valuable measure. Additionally, precision and recall will also be considered. Recall represents the number of correctly classified positive events from all positive events. Precision represents the number of correctly classified positive events from all positively classified events (see the formulas below). The F1 score represents a way to combine precision and recall into one measure. Logloss uses the predicted probability of the model instead of the target variable (0.93 instead of 1). Logloss is known for heavily penalizing wrong predictions. Additionally, the time it takes for each model to train and predict are also included. Finally, the extent to which each model is transparent or a ‘black-box’ is also included. This is not a calculated metric, but rather additional information that provides a more complete comparison of the different models. For deep learning models the epoch at which the highest accuracy was achieved is also recorded.

Table 8

Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

The evaluation metrics can then be formulated as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$

4. Results

This chapter will present the results of the analysis. First, descriptive statistics of the dataset will be presented. Second, the traditional machine learning models will be compared on the various metrics presented in the previous chapter. Third, the deep learning models will also be compared on these metrics. Finally, the best performing models will be tuned through hyperparameter optimization.

4.1 Descriptive Statistics

This section will present descriptive statistics on the datasets used for both type of models. First, the descriptive statistics for the traditional machine learning models are discussed. Next, descriptive statistics on the sequence of user behavior used for the deep learning models are discussed.

4.1.1 Traditional Machine Learning Models

Descriptive statistics for all numerical features can be found in table 9, while table 10 presents information on all categorical features. The dataset (for both machine learning and deep learning models) consists of 75.566 users. The dataset is completely balanced and consists of 36.283 users that converted and 36.283 users that did not convert. Descriptive statistics presented in tables 9 and 10 are calculated separately for the two groups, converting users and non-converting users. This way the relationship between each feature and the target variable can be explored. On average, converting users report a higher mean value for main interactions like the total number of interactions, the number of page views and the number of page pings. Converting users also show much more scrolling (20.656 vs 11.984). This could indicate that converting users interact more with the website than non-converting users. On the other hand, non-converting users do more personality tests (0.11 vs 0.03), visit more homepages (0.92 vs 0.39), more account pages (0.31 vs 0.13) and more scholarship pages (0.37 vs. 0).

When not taking the target variable into account, the average number of interactions per user is 34.65. On average, a user experiences more page ping events (14.67) than page view events (6.28). This falls in line with expectations as it is expected that a user would scroll more with a page than view pages. Table 9 also shows that the scrolling feature is on a different scale than the other features. The tree-based machine learning models know how to handle this difference. Other machine learning models, like k-Nearest Neighbors cannot naturally handle this which is why the features are normalized for these models (see section 3.3.1). The mean of both the country test and the personality test features are low. These features are measured as binary variables. Especially the country test does not seem to be used by many users. Additionally, much more non-converting users than converting users did a personality test. Table 10 also shows that many non-converting users entered StudyPortals on the personality test page. Exploring this actual page showed why the personality test generally leads to non-converting users. The page presents the user with a test and the user is afterwards informed that the results will be send by e-mail. However, after completing the test there is no easy way to go back to the other portals on the StudyPortals website. The target variable has a mean of 0.5 and a standard deviation of 0.5. This is as expected because the dataset is balanced with half of the dataset consisting of converting users (1) and the other half consisting of non-converting users (0).

Table 10 shows that the feature ‘geo_country’ has the most unique values with 221, while ‘device_type’ has the least with 5. Most users enter the website on www.mastersportal.eu. BachelorsPortal is the second most popular portal. Most non-converting users are from India and there is quite a difference for Indian users when it comes to whether they converted or not. The distribution of browser families is very similar for both groups with Google Chrome being the most used browser. Most converting users

use a computer and have Windows as their operating system. For non-converting users the mobile device with Android operating system is much more used than by converting users.

Table 9

Mean values of numerical features

Feature	Non-converting users	Converting users
Total interactions		
No_of_interactions	30.67	38.64
Online tests		
No_of_ctests	0.001	0.001
No_of_ptests	0.11	0.03
Page pings		
Page_pings	14.11	15.21
Scrolling	11984.07	20656.01
Page views		
Page_views	6.07	6.51
Avg_page_time	310.72	37.89
No_studies	1.08	2.68
No_searches	0.96	1.33
No_homepages	0.92	0.39
No_study_options	0.58	0.53
No_universities	0.44	0.39
No_disciplines	0.38	0.52
No_articles	0.28	0.11
No_countries	0.33	0.34
No_account	0.31	0.13
No_scholarships	0.37	0
Target		
Target	0	1

Table 10

Descriptive statistics of categorical features

		Non-converting users		Converting users	
Feature	Unique values	Value	%	Value	%
Page_urlhost	15	MastersPortal	42.16	MastersPortal	58.99
		BachelorsPortal	13.27	ShortCoursesPortal	17.04
		Personalitytest	9.52	BachelorsPortal	13.57
Geo_country	221	India	15.43	India	10.73
		United States	7.85	United Kingdom	7.32
		United Kingdom	6.44	United States	7.04
Br-family	10	Chrome	63.73	Chrome	66.29
		Safari	15.14	Safari	14.81
		Firefox	9.66	Firefox	11.34
Os_family	11	Windows	48.44	Windows	60.49
		Android	28.80	Android	14.48
		IOs	11.38	Mac OS X	13.00
Dvce_type	5	Computer	58.23	Computer	74.59
		Mobile	37.42	Mobile	21.42
		Tablet	3.80	Tablet	3.89

4.1.2 Deep Learning Models

Table 11 presents descriptive statistics on the behavioral sequence used for the deep learning models. The total number of behaviors for all users is 1.627.866. The maximum number of behaviors per user is 6511, while the minimum is 3. This is because only users with at least 3 page views are included in the dataset. The dataset contains 187 unique behaviors. The average number of behaviors per users is 21.26, where converting users have an average of 0.84 behaviors more than non-converting users. As described in the methodology chapter the sequence for each user gets padded until it has a length of 125 behaviors. The two most occurring behaviors for both converting and non-converting users are a page ping event where zero scrolling happens and a page view on a study page on MastersPortal. Zero scrolling occurs when a page ping gets logged but the user did not scroll in between page ping events. The third most occurring behavior for both classes is a page ping event with different amounts of scrolling. These most occurring behaviors are very similar. This stresses the importance of not only considering how many times a behavior appears, but also to consider in what order they appear.

Table 11

Descriptive statistics on the top 3 most occurring user behaviors

	Converting users			Non-converting users		
Most occurring behavior	Index	Behavior	%	Index	Behavior	%
#1	6	Page ping: 0	13.95	6	Page ping: 0	20.36
#2	4	Page view: MastersPortal study page	8.34	4	Page view: MastersPortal study page	3.51
#3	15	Page ping: 301-400	3.71	2	Page ping: 101-200	3.45

4.1.3 Correlations

Figure 9 represents an overview of the correlations between the numerical features used for the traditional machine learning models. Because the data is not normally distributed, Spearman's rho has been used.

The highest positive correlations are between the number of page pings and the number of interactions (r = .89) and between the number of page views and the number of interactions (r = .79). This is expected as the number of interactions is simply the sum of the number of page views and the number of page pings. Additionally, there also is a high positive correlation between the number of study pages viewed and the number of total interactions (r = .71). The highest negative correlation is between the number of study option pages visited and the number of home pages visited (r = -.17). There also is a negative correlation between the personality test variable and the target variable (r = -.16). As mentioned before the page that host the personality test leads to a 'dead-end'.

When looking at the correlations between the features and the target variable the highest positive correlation is between the number of study pages visited and the target variable (r = .23). This is expected as the button to convert is only found on study pages. The highest negative correlations are found between the target variable and whether a user did a personality test (r = -.16), the number of viewed homepages (r = -.16) and the number of viewed scholarship pages (r= -.11).

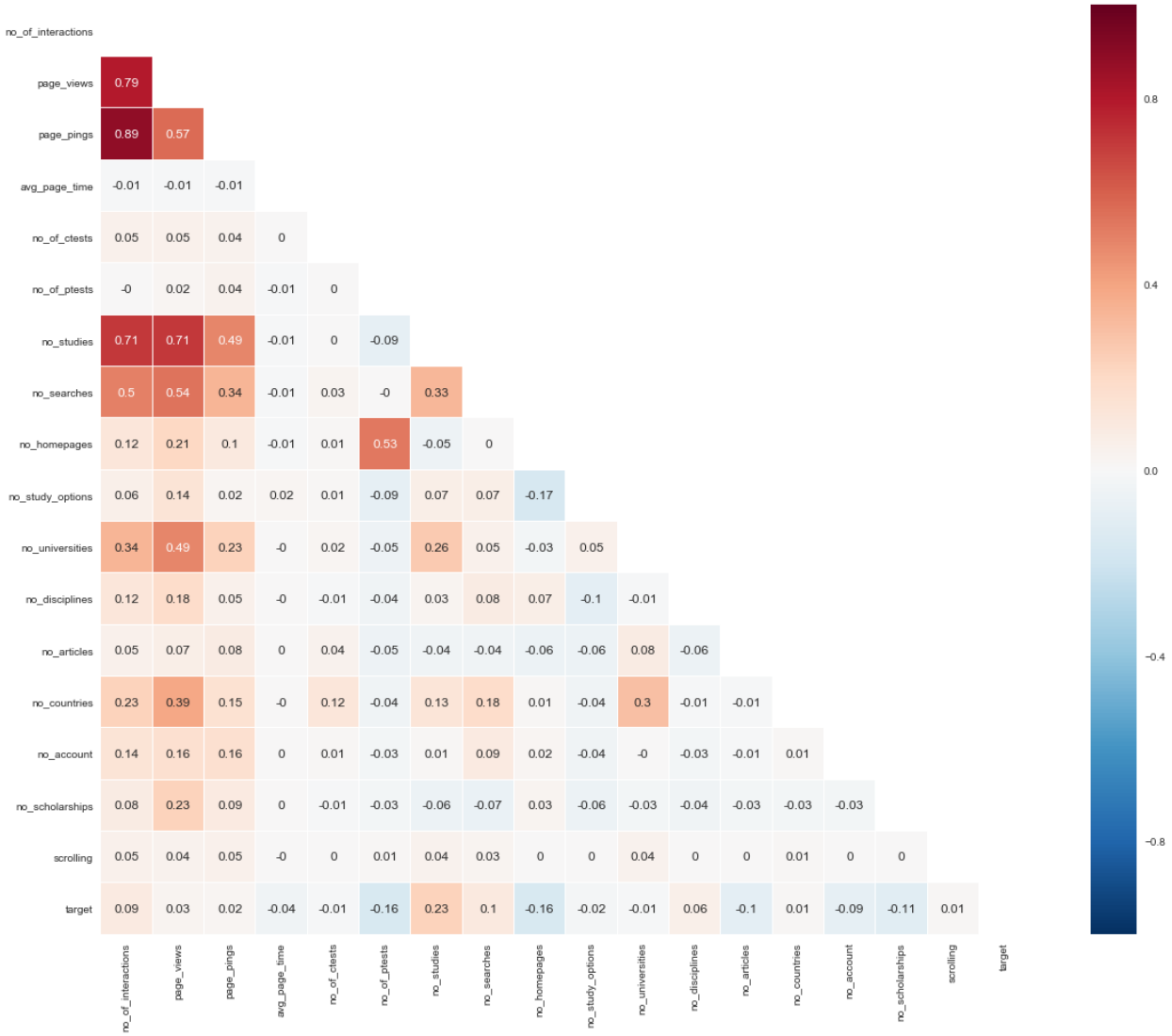

Figure 9. Correlation matrix.

4.2 Traditional Machine Learning Models

This section presents the findings of comparing the traditional machine learning models. An overview of this comparison can be found in table 12. First, the accuracy of all models is compared. Accuracy is the clearest metric and provides a quick way to compare different models. The SVM with rbf kernel shows the worst accuracy, followed by the k-NN algorithm with 66.80% and 69.76% respectively. The tree-based models show the highest accuracies. The Random Forest (RF) and the Gradient Boosted Trees (GBT) models are ensembles of decision trees and expectedly perform better than the single decision tree. Both the RF and the GBT are the only models to score over 80% accuracy with 80.60% and 82.20% respectively. Because the dataset consists of a balanced dataset it is expected that the precision, recall and f1-score are similar for each model. Table 12 shows that these three scores are similar for all models except for the GBT model. The GBT model shows very high precision with 0.8553. The precision for this model is higher than the recall score of 0.8220. The confusion matrix of the model shows that the GBT model predicts class 1 more often than it predicts class 0. Additionally, there are very few cases where it predicted class 0 and got it wrong. There seems to be a general trend that precision is slightly higher than recall for all models. When exploring logloss we can see that the decision tree scores much higher than the other models. This is caused by the fact that a decision tree is not made to predict probabilities. The end of the tree results in a class prediction. Logloss is a metric that the model tries to minimize so a higher Logloss indicates a worse performance. The Logloss for Logistic Regression and the RF are similar despite their difference in accuracy being much bigger. The GBT model shows the lowest logloss with 0.3619.

There are two machine learning models that show high training and prediction times. These models are k-NN and the SVM with rbf kernel with combined training and prediction times of 423.63 and 1265.63 seconds. All other models take less than 10 seconds to do the training and prediction. The training takes place on 70% of the dataset and prediction on the other 30%. k-NN is the only model where prediction takes longer than training. This is because the k-NN algorithm has to scan through the whole dataset for each prediction it makes to find the nearest neighbors. LR, SVM (linear kernel), DT, RF and GBT models all predict very fast. Therefore, all these models would be potential candidates for a system focused on real-time prediction.

Finally, depending on the goal of the prediction it can be important to what extent the model is transparent. Some machine learning models operate like a ‘black-box’ where it is almost impossible to see what features are important and how the model comes to its final prediction. Logistic Regression and single Decision Trees are very transparent methods. In Logistic Regression, the beta values show the magnitude of each feature while for a decision tree the full tree can be printed out. You can then follow along the tree and see where each prediction would end up. On the other hand, Support Vector Machines are complete black boxes. K-NN, Random Forest and Gradient Boosted Trees fall somewhere in the middle. For RF and GBT we can explore different trees in the ‘forest’ and get a sense of how each tree is build. For both these models the importance of each feature can be calculated using the Gini importance (Breiman, Friedman, Stone & Olshen, 1984).

Based on all metrics the Random Forest and the Gradient Boosted Trees models perform best. These models score the highest on accuracy, precision, recall and f-1 score. Additionally, these models are also quick at prediction and fairly transparent. Therefore, these two models will be tuned through hyperparameter optimization. Section 4.4 will show how much their performance can be improved through this process.

Table 12

Comparison of Traditional Machine Learning models on various metrics

Model	Accuracy	Precision	Recall	F1-score	Logloss	Training Time	Prediction Time	Epoch	TP
Logistic Regression	0.7366	0.7467	0.7366	0.7340	0.6357	2.78	0.05	NA	H
k-Nearest Neighbors	0.6976	0.7048	0.6976	0.6950	1.93	9.99	413.64	NA	M
SVM (linear)	0.7215	0.7376	0.7215	0.7295	0.9545	2.44	0.04	NA	L
SVM (rbf)	0.6680	0.7012	0.6732	0.6869	3.4925	1011.30	254.33	NA	L
Decision Tree	0.7689	0.7689	0.7689	0.7689	7.98	0.39	0	NA	H
Random Forest	0.8060	0.8085	0.8060	0.8057	0.7049	0.65	0.07	NA	M
GBT	0.8220	0.8553	0.8220	0.8179	0.3619	4.60	0.05	NA	M

Note. GBT = Gradient Boosted Trees. H = highly transparent. L = Not transparent, like a black-box. M = Transparent to a degree. SVM = Support Vector Machine. TP = Transparency

4.3 Deep Learning models

Table 13 presents an overview of the comparison on deep learning models. All these models perform better than the traditional machine learning models. Additionally, the metrics for these deep learning models are much closer to each other than they were in the comparison of the traditional machine learning models. On accuracy, the RNNs with LSTM and GRU layers perform best with 89.82% and 89.86% respectively. The simple neural network with 1 hidden layer performs worst, but still scores 88.38% accuracy. It should be noted that technically this is not a 'deep' learning model since it only has one hidden layer. The scores for precision, recall and the f1-score are similar for each model. Like the traditional machine learning models, precision is slightly higher than recall for all models. The RNNs with LSTM and GRU layers also perform best when using the logloss metric.

The training times and prediction times are given per epoch. As described in the methodology chapter all models are run for ten epochs and metrics are reported for the best epoch. Most of the traditional machine learning models could be trained and do the prediction in under ten seconds. For the deep learning models, only the deep neural networks are this fast, other architectures are not. The RNNs are the slowest models because for these models each input is entered sequentially which makes parallelization much harder. The simple neural networks are the only architectures were the prediction time is fast enough to allow for a kind of real-time prediction. The more complicated architectures seem to achieve their best result at later epochs. For example, the RNNs with LSTM and GRU layers do not reach their best accuracy until epoch nine. Finally, all deep learning models score similar when it comes to transparency. Essentially they are all 'black-boxes'. It is very difficult to find what features are important and why. One of the only options is to do some kind of sensitivity analysis and exclude features one-by-one and explore the effect on the model's performance. Current research is exploring different ways to make deep learning models more transparent but they are not there yet.

Table 13

Comparison of Deep Learning models on various metrics

Model	Accuracy	Precision	Recall	F1-score	Logloss	Training Time	Prediction Time	Epoch	TP
NN with 1 HL	0.8838	0.8892	0.8838	0.8834	0.2892	4.48	0.66	5	L
NN with 3 HL	0.8850	0.8908	0.8850	0.8846	0.2884	5.31	1.05	3	L
CNN with 1 CL	0.8941	0.9001	0.8941	0.8938	0.2659	16.44	4.42	5	L
CNN with 2 CL	0.8967	0.9017	0.8967	0.8964	0.2654	24.53	7.57	7	L
RNN	0.8918	0.8961	0.8918	0.8916	0.2758	48.58	18.46	3	L
RNN LSTM	0.8982	0.9040	0.8982	0.8978	0.2533	91.44	27.05	9	L
RNN GRU	0.8986	0.9035	0.8986	0.8984	0.2549	159.82	29.26	9	L

Note. CL = Convolutional Layer. CNN = Convolutional Neural Network. GRU = Gated Recurrent Unit. H = highly transparent. HL = Hidden Layer. L = Not transparent, like a black-box. LSTM = Long-Short Term Memory. M = Transparent to a degree. NN = Neural Network. RNN = Recurrent Neural Network. TP = Transparency.

4.4 Hyperparameter Optimization

In this section, the hyperparameters of the best performing models of section 4.2 and 4.3 will be optimized. For the traditional machine learning models the best performing models were the Random Forest and Gradient Boosted Trees models. The best performing deep learning models were the Recurrent Neural Network with Gated Recurrent Units and the Recurrent Neural Network with Long-Short Term Memory. All Machine learning models have a set of parameters that are learned from the training data. Additionally, most models also have other parameters that cannot be learned from this training process. These hyperparameters have to be set by the researcher and present 'higher-level' parameters. In the previous section the models were compared while using the base values of the hyperparameters. In this section, the hyperparameters will be tuned to find the optimal values for them. Then, the improvement of these models through hyperparameter optimization can be explored.

4.4.1 Random Forest

The process of hyperparameter optimization is also referred to as 'tuning'. The first step to tuning a Random Forest model is to decide how many decision trees should be used in the tuning process. Using many trees would make the tuning process very expensive to compute and might make the model too prone to overfitting. Figure 10 presents the accuracy score per number of decision trees in the random forest. The best accuracy score is achieved by using 500 trees. However, using 250 trees provides an accuracy that is very similar and computationally more efficient. Therefore, 250 trees will be used during the hyperparameter optimization process.

The Random Forest algorithm has few hyperparameters too tune. This makes the model an easy choice for quickly testing a machine learning model. The hyperparameters that were tuned are: the maximum number of features to consider at each split in the three and the maximum depth of each tree. A grid search showed that the optimal values are 8 for the maximum number of features and 15 for the maximum depth. Re-training the Random Forest model with 500 trees and the other optimium hyperparameters provides an accuracy of 83.04%. Optimizing the hyperparameters has increased the accuracy from 80.60% to 83.04%. Figure 11 shows the feature importance plot of the Random Forest

model. It can be seen that the model places a great importance on the number of study pages visited by the user. The number of total interactions, average time spent on each page, the number of searches performed and the sum of vertical scrolling also are important features in the RF model. On the other hand, the personality test and country test features are not seen as important by the RF model.

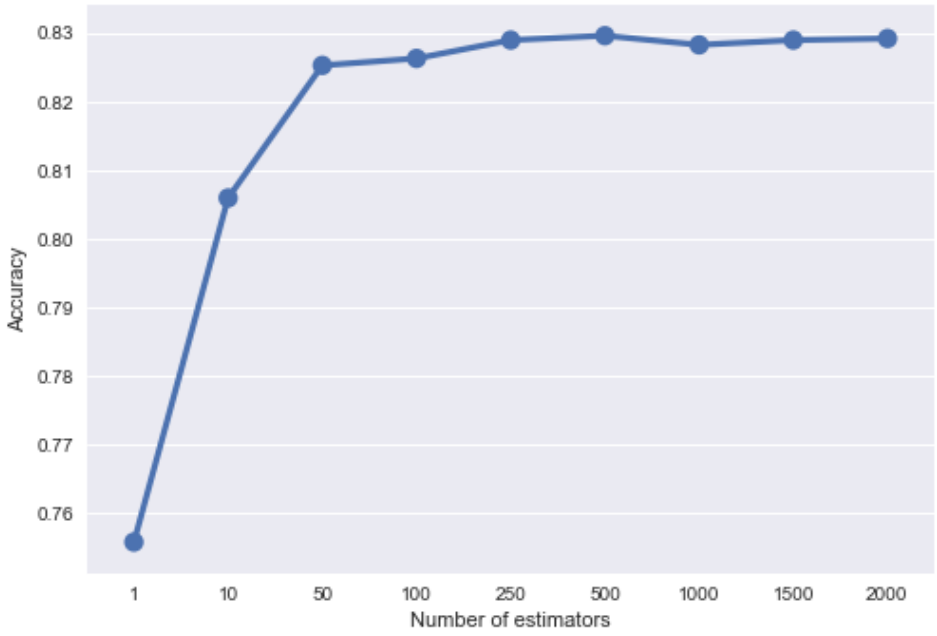

Figure 10. Accuracy per number of estimators (trees) in the Random Forest model.

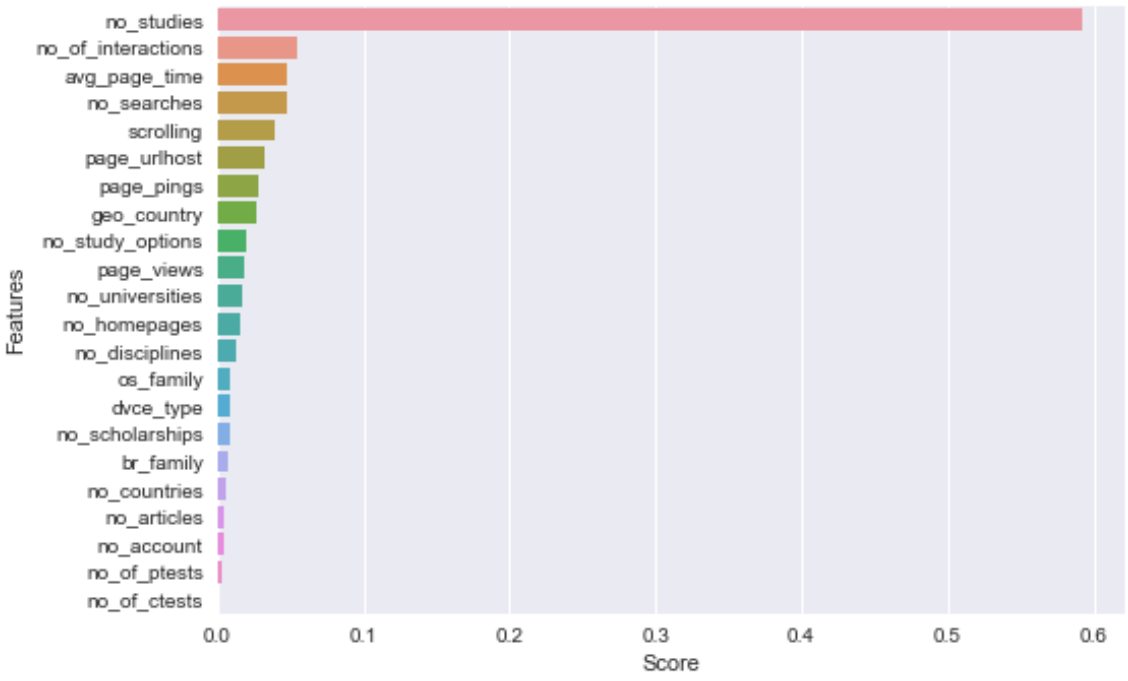

Figure 11. Feature Importance of the Random Forest model

4.4.2 Gradient Boosted Trees

Compared to the Random Forest, the Gradient Boosted Trees model has much more hyperparameters too tune. Therefore, it is even more important to choose the right number of trees used for hyperparameter optimization. To be sure of the number of trees to use, we run a function that trains the model with 10-fold cross validation. This finds the number of trees to use and stops adding trees after accuracy on the test set has not improved after adding 50 additional trees. According to this function, 274 trees is shown as the number of trees to use for tuning. Figure 12 shows how the logloss and classification errors evolve on the training and test set when using up to 2000 trees. Figure 12 also shows that after this point the model starts to overfit on the training set and therefore starts performing worse on the test set.

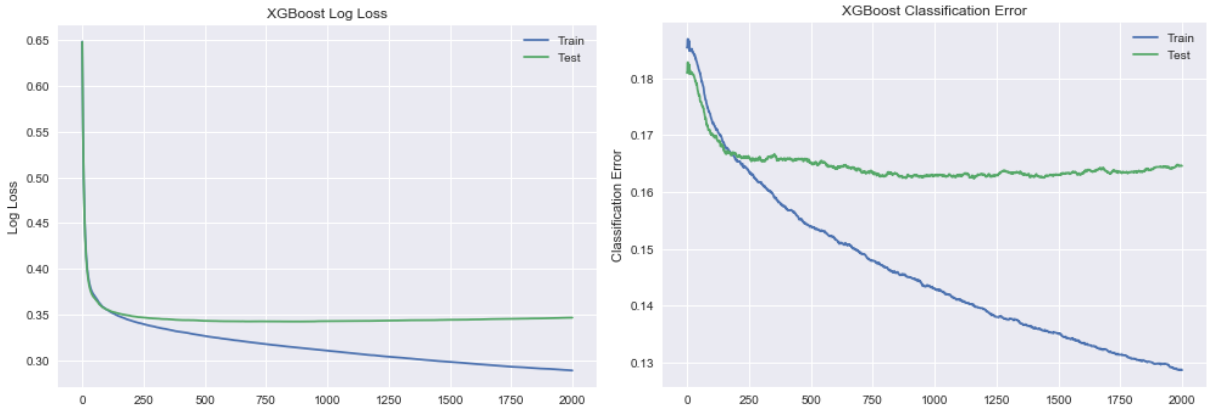

Figure 12. Log loss and classification error per number of trees in the Gradient Boosted Trees model.

Tuning all hyperparameters is a heavy process because the model must be trained and tested for every combination of parameters. Additionally, the grid search used for testing the combinations of parameters uses cross-validation which means that every combination will be calculated multiple times. The optimal values for each hyperparameter can be found in in table 14.

Table 14

Hyperparameters and their optimal values for the Gradient Boosted Trees model

Hyperparameter	Optimal value
Number of trees	274
Maximum depth of tree	7
Minimum child weight	5
Gamma, minimum loss reduction	0.8
Subsample ratio for training	0.8
Subsample of columns to use for each tree	0.8
L1 regularization	0.1
L2 regularization	50

Re-training the Gradient Boosted Trees model with these parameters increases its accuracy from 82.20% to 83.61%. Thus, the GBT model still performs better than the RF model. Although, the difference in performance between the two models has become smaller. Additionally, the RF model needed much less tuning than the GBT model. Figure 13 presents the importance of each feature in the GBT classifier based on their gini importance. Scrolling is the most important feature. Next, the average page time, country of the user and the total number of interactions all have a similar importance in predicting conversion. Section 4.1 already showed big differences in the amount of scrolling and average page time for the different user groups. The number of scholarship pages visited and whether a user did a personality test or country test have the least importance in predicting conversion with this model. Figures 11 and 13 show that the important features are quite different for the RF and GBT models. Despite placing different levels of importance on each feature the models achieve similar accuracies (83.04% and 83.61%).

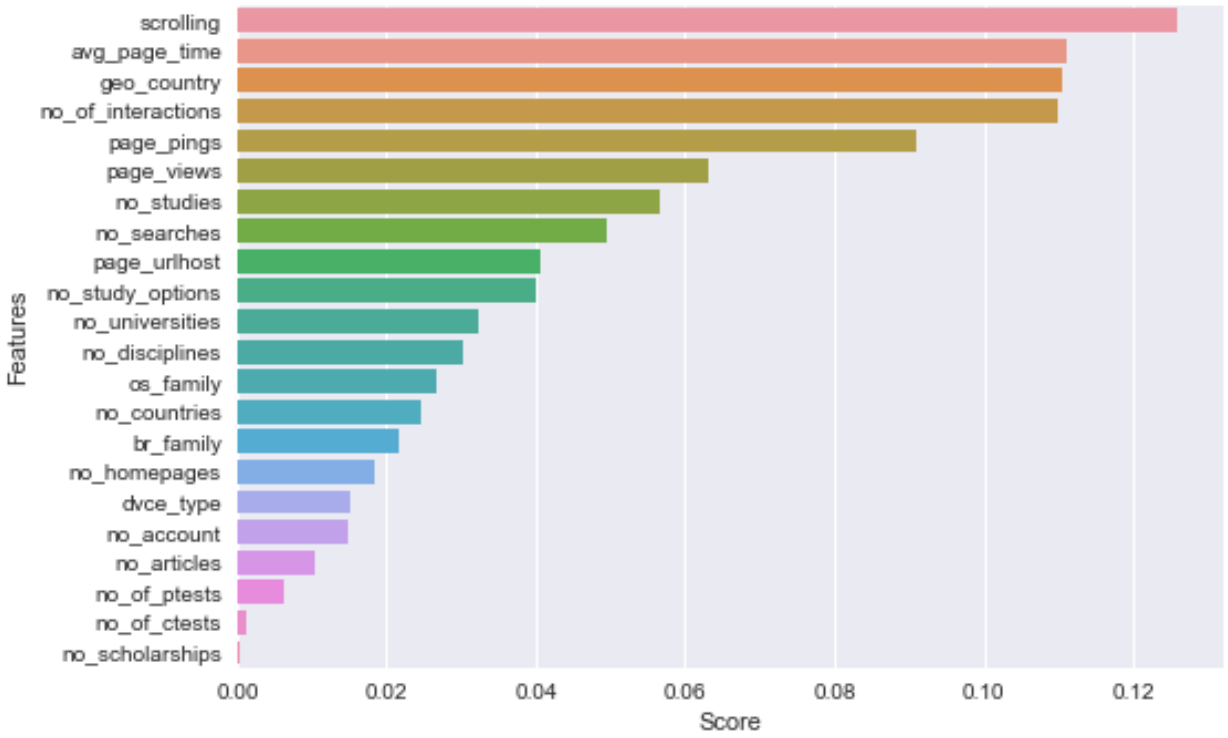

Figure 13. Feature Importance of the Gradient Boosted Trees model.

4.4.3 Recurrent Neural Network with LSTM and GRU

The Recurrent Neural Network with LSTM and the Recurrent Neural Network with GRU have been tuned on the following hyperparameters:

- The optimizer used
- The number of hidden units in the recurrent layer
- The size of the embedding layer

This tuning process is computationally expensive and takes a long time to run. Choosing different values for these hyperparameters showed to have little impact on the performance of each model. The RNN with LSTM improved its accuracy from 89.82% to 90% with optimal values for the hyperparameters. The RNN with GRU improved its accuracy from 89.86% to 89.98%. The optimal combination of parameters for both models was: Adam optimizer (learning rate = 0.001), 50 hidden units and 50 latent factors in the embedding layer. Table 15 presents an overview of the increased performance of all models from this section.

Table 15

Overview of Machine Learning models before and after hyperparameter optimization

Model	Accuracy before (%)	Accuracy after (%)	Description of tuning process
RF	80.60%	83.04%	Simple and straightforward
GBT	82.20%	83.61%	Many parameters to tune
RNN - LSTM	89.82%	90.00%	Computationally very expensive
RNN - GRU	89.86%	89.98%%	Computationally very expensive

Note. GBT = Gradient Boosted Trees. GRU = Gated Recurrent Unit. LSTM = Long-Short Term Memory. RF = Random Forest. RNN = Recurrent Neural Network.

5. Conclusion

This section will present the main conclusions of the research and will answer the research problem. Additionally, limitations of the research will be discussed and suggestions for future research are presented. The sub questions have already been answered throughout this research. Therefore, this section will first provide a short summary of the answer to each sub question. Afterwards, the research problem will be answered.

What variables has previous literature identified as being significantly related to conversion?
Previous literature has identified the number of page views and session length as significant predictors of conversion (Bellman et al., 1999; Lin et al., 2010). Later, research into conversion started to switch towards more complicated variables. Recently, research into predicting conversion has also identified the different page types on a website, scrolling and the sequence of user behavior to significantly influence conversion (Goldstein et al., 2017; Guo & Agichtein, 2016; Lo et al., 2016).

What are relevant metrics in the comparison of traditional machine learning and deep learning models?
In general the main metric used will have to be decided for each specific project. However, this study did show the importance of including multiple metrics. Where one model might score high on accuracy it might take very long to train, which makes it not practical to use. Additionally, metrics like recall, precision and the F1-score can give a better estimate of performance than accuracy when dealing with unbalanced datasets. Especially in practical applications the small percentage points in accuracy gain are often not worth the downsides that come along with certain models. Therefore, it is crucial to include a combination of metrics when testing machine learning models. Deep learning models are computationally expensive to train and to tune. Therefore, it can be even more important to take into account training and prediction times when working with deep learning. In this study a comprehensive combination of metrics was used consisting of accuracy, precision, recall, f1-score, Logloss, training time, prediction time, the best performing epoch and the transparency of the model. This combination of metrics provides a more complete assessment of the performance of each model. By combining these metrics one can take into account both the predictive quality and the practical usability of the model.

What pre-processing steps should be taken in order to compare traditional machine learning and deep learning models?
When comparing traditional machine learning models it is important to decide how categorical variables are handled. Label encoding and one-hot encoding are the most popular choices for pre-processing categorical variables. Additionally, normalization of numerical features might have to be applied depending on the traditional machine learning model being used. An experiment in the methodology chapter showed the importance of selecting the right pre-processing tasks for each model. It showed, for example, that a model that is known to require a normalized input decreases in accuracy when it receives non-normalized input. When using an embedding layer for the deep learning models, the values in that vector are already normalized from the start. Additionally, embedding layers provide another way of handling categorical data. Embedding layers have the main advantage that they do not produce big spare vectors like one-hot encoding does when there are many levels for a categorical variable. When using sequential data in deep learning one must decide on the sequence length. Because deep learning does not require explicit feature extraction the overall process of going from raw data to pre-processed data is often much shorter than it is for traditional machine learning models.

Do deep learning models perform better in predicting customer conversion than traditional machine learning models?
In this study all deep learning models performed better than the traditional machine learning models when looking at the quality of the predictions. Deep learning is able to take into account the sequential aspect of the data. This is one of the main advantages of deep learning and therefore it was also used in this study. If you were to input the exact same tabular data format into deep learning models as into the traditional machine learning models, then not all deep learning models would perform better. In conclusion, it is important to understand what data you are dealing with and also whether that sequential aspect of that data will likely have an impact. From this case study it seems that if your data has a sequential aspect, then deep learning will likely outperform traditional machine learning on metrics like prediction accuracy. On the other hand, when taking into account metrics related to practical usability of deep learning models they perform less good than traditional machine learning models. Prediction times and training times of deep learning models are longer, they are less transparent and they need to be trained GPUs instead of more common CPUs. However, when looking at the prediction accuracy, deep learning models perform better than traditional machine learning models.

After answering the sub questions the research problem can now be answered.

Research problem: *What is the value of deep learning models for predicting customer conversion?*

Before focusing on the value of deep learning models it is first important to reflect on some of the disadvantages of using deep learning. First, deep learning models are essentially black-boxes. It is difficult to get an understanding of what features are most important for the model in getting to its final prediction. There are techniques that can be used to, for example, visualize what filters a convolutional neural network is learning. However, these techniques are time consuming and have their own limitations. In some cases of conversion prediction it might be crucial to understand the most important features in the prediction model. Second, big amounts of data are needed before deep learning models can perform optimally. While traditional machine learning models are known to have their performance stagnate at a certain data size, deep learning models keep getting better with the more data they get. Third, because deep learning deals with big data and complex matrix operations it cannot be run locally on most computers. Essentially, a GPU designed for deep learning is necessary to run deep learning models on a reasonably sized dataset. Either investments into GPUs have to be made or external servers can be used for this process. Basically, the use of GPUs requires more knowledge and costs than traditional machine learning models. These costs are of monetary value as GPU servers can be expensive, but also cover time investments. Deep learning models often take longer to train, validate and tune than other machine learning models. Because deep learning models take longer to do predictions on new data, this also means that many deep learning architectures are not suitable for real-time production. There are solutions to this by using distributed computing systems like the Hadoop ecosystem. However, these are costly to run and are not a solution for smaller companies or researchers.

However, there also are many advantages to the use of deep learning models when predicting conversion. First, all deep learning models achieved better prediction performance than the traditional machine learning models. Especially on metrics like accuracy, f1-score and Logloss there is a big improvement in using deep learning models versus the use of traditional machine learning models. Even after carefully tuning the traditional machine learning models, the best performing deep learning model

still scored over six percent higher in accuracy. An accuracy gain from 83.61 to 90.00 is a big gain. The higher the accuracy score of a previous model, the more difficult it is to increase accuracy over that already high score. Deep learning models clearly performed better at predicting conversion in this study than the traditional machine learning models. Second, deep learning models are able to capture sequential relationships in data. When prediction conversion most datasets will be click-stream data that have this sequential element. Deep learning models can take advantage of this and incorporate this element in their prediction. Third, deep learning models are able to capture very complex relationships and patterns. For example, a deep neural networks consists of multiple hidden layers and as the network gets deeper it starts to capture increasingly complex and non-linear patterns. These patterns might be difficult and for some machine learning models impossible to find. For example, Recurrent Neural Networks have internal memory gates that allow the network to make its own decisions about what data it should keep in memory and what it should forget. Concluding, the value of using deep learning models is found in its better predictive performance. As datasets become bigger and bigger deep learning is able to use this increase in available data to its advantage. Where the performance of traditional machine learning models stagnates, deep learning models' performance keeps increasing. Value is also found by how deep learning can capture complex relationships and sequential patterns in datasets that traditional machine learning models cannot capture. Because of this, deep learning models outperform traditional machine learning models greatly when it comes to the predictive accuracy of the model.

What does this mean for the marketing industry and computer scientists? Traditional machine learning models are still useful in certain cases. For example, they function as a tool to quickly prototype a prediction model. Additionally, in most cases they also offer transparency into how the model makes a prediction and what features are most important. Ensemble methods like Random Forest and Gradient Boosted Trees still perform better on tabular data and smaller datasets than deep learning models. Thus, when any of the above is important to the research it is advisable to stick to traditional machine learning models. However, deep learning adds huge value when the data has a sequential element, like images, audio, clickstream data and language. The predictive performance of deep learning models is so much better than that of traditional machine learning models in these cases that practitioners should try to implement deep learning in these cases. Deep learning models need to be trained on a GPU. Currently, GPUs are becoming cheaper and more easily available. Simultaneously, there are many new deep learning libraries and tools released every month. Therefore, the adaptability of deep learning keeps increasing and it will become available for everyone with programming experience in the near future.

Research has already explored the use of deep learning in predicting the next step of visitors on a website and achieved positive results with this (Tamhane et al., 2017; Tang et al., 2016). These results suggest that there is a potential of deep learning in marketing and that the hype surrounding deep learning seems justified. Previous marketing research already mentioned that more complicated models were necessary for predicting conversion and purchases (Goldstein et al., 2017; Lo et al., 2016). Deep learning provides these models that can capture complex and non-linear relationships. Next to predicting conversion, purchases and churn there are also other applications of deep learning in marketing. Deep learning can also be used for unsupervised learning. This is a type of machine learning where the target variable is unknown. In these cases deep learning could be used for client segmentation. However, more research is needed to see how deep learning performs against clustering algorithms like k-means. This study specifically focused on the task of predicting conversion. However, we believe that these results do not only count for the specific issue of predicting conversion, but can also work for other marketing cases such as predicting churn, brand awareness and the type of

persuasion technique to use. It all depends on the data that is available to train the deep learning model on. If clickstream data is available it is fairly easy to extend these models to predict different things. For example, let us imagine that we used this dataset to predict churn instead of conversion. From the clickstream dataset used in this study we could find when and if each user returned to the website. Then all we have to do is change the target variable from whether the user converted or not to whether the user returned or not. Additionally, this could easily be turned into a multi-classification where more than two classes can be predicted. Fore example, whether the user did not return, whether they returned within less than three days or within more than three days. Because deep learning does not require the extensive feature extraction step that traditional machine learning requires, we can construct the sequence of user behavior in the same way as was done in this study. Then all there is left to do is retrain the deep learning models on this new dataset. This shows that these models for predicting conversion can easily be extended to predict churn. Of course we do not know exactly how well these models would perform. Although, it is expected that they would perform similarly due to how well deep learning models are at capturing complex patterns in the dataset.

5.1 Limitations

This section will discuss the limitations of the study. First, this study has been conducted with data for one marketing problem with data from one company. Although it seems reasonable to expect that click-stream data from different organizations and industries would lead to similar results this cannot be taken for granted. Therefore, a degree of carefulness is necessary when applying these results on different datasets that might not share many resemblances to this dataset. Second, because the datasets for both types of models have been pre-processed in different ways it can be difficult to compare the results of the two approaches. On one hand, the exact same dataset has been used to train both types of models. On the other hand, through feature extraction and pre-processing the final input that goes into these models is different for both types of models. Of course this is necessary since both types of models perform best on a different input, but it might influence the comparison. This means that the results should be interpreted when comparing traditional machine learning and deep learning overall. One should not focus too much on the comparison of two specific models from the different groups, like comparing Support Vector Machines and Convolutional Neural Networks. An issue for future research would be too study if there might be a different approach to comparing all these different models. Third, this research addressed many different factors and metrics but it did not address changes in the size of the dataset. It is a general rule that deep learning becomes more useful than other machine learning models as the size of the dataset increases. It could have been interesting to repeat the same experiments done in this study on different sizes of the dataset. Then, we could see at what point it might be useful to go for traditional machine learning models and at what data size deep learning models start performing better. However, this would have still been affected by the type of data used in this study and the results might not have been generalizable.

5.2 Future Research

Based on the findings of this research the following suggestions for future search have been formulated. First, we suggest that future research focuses more on empirical research of deep learning applications in marketing. As mentioned at the start of this research, there are many theoretical papers that discuss the potential of deep learning in marketing. However, there are not many empirical papers testing and validating this potential. This research acts as a starting point for these type of studies. This study has shown that deep learning performs well in predicting conversion. The deep learning models perform much better than the traditional machine learning models on prediction accuracy, which shows that there is great potential for deep learning in marketing. Churn prediction, client segmentation, purchase prediction, personalization on e-commerce platforms are all marketing areas where deep learning can have valuable contributions.

Second, future research could look at a robust method for comparing different machine learning models. This study also showed that it is important to, for each machine learning model, pre-process the data in a way that the model expects. Other researches in the area of educational data mining do not discuss this in detail. It is therefore unclear if some researchers did apply this method of pre-processing or if they simply put all the same data into different machine learning models. Additionally, it can be hard to compare the results between traditional machine learning and deep learning models because most deep learning models expect a different input. Therefore, we suggest that future research tries to come up with a methodology for comparing all these different models.

Third, in finding this methodology for comparing traditional machine learning and deep learning models we suggest future research to specifically focus on the metrics used. This study combined a set of metrics related to the predictive performance and practical usability of the models. We hope that this can be a starting point towards a quality comparison model. This would be a type of standardized model that all research can use when comparing deep learning models with traditional machine learning models. Perhaps a machine learning model could be trained to learn what metrics are most relevant in these comparisons.

Finally, we suggest that future research looks into the possibilities of using more complicated deep learning architectures. To provide an accurate comparison the architectures of the deep learning models were kept fairly straightforward in this study. We did experiment with different architectures. For example, an architecture that started with a Recurrent Neural Network that takes the sequence of user behavior and later on adds the metadata of the user and puts this through another deep neural network. However, we found that his approach did not change the prediction accuracy of the model much while it did make the model much slower. In order to keep all comparisons as fair as possible these models were kept out of this study. However, we would like to see future research explore more exotic architectures of deep learning in marketing. Hopefully, that research can find specific architectures that perform well for specific marketing solutions.

6. References

Alper, M. E., & Cataltepe, Z. (2012). Improving Course Success Prediction using ABET Course Outcomes and Grades. In CSEDU (2) (pp. 222-229).

Barber, M., Donnelly, K., Rizvi, S., & Summers, L. (2013). An avalanche is coming: Higher education and the revolution ahead. Institute for Public Policy Research, 11.

Bellman, S., Lohse, G. L., & Johnson, E. J. (1999). Predictors of online buying behavior. Communications of the ACM, 42(12), 32-38.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In Neural networks: Tricks of the trade (pp. 437-478). Springer Berlin Heidelberg.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

Bucklin, R. E., Lattin, J. M., Ansari, A., Gupta, S., Bell, D., Coupey, E., ... & Steckel, J. (2002). Choice and the Internet: From clickstream to research stream. Marketing Letters, 13(3), 245-258.

Chan, T., Joseph, I., Macasaet, C., Kang, D., Hardy, R. M., Ruiz, C., ... & Honda, T. (2014, June). Predictive Models for Determining If and When to Display Online Lead Forms. In AAAI (pp. 2882-2889).

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., & Andrew, N. (2013, February). Deep learning with COTS HPC systems. In International Conference on Machine Learning (pp. 1337-1345).

Dekker, G., Pechenizkiy, M., & Vleeshouwers, J. (2009, July). Predicting students drop out: A case study. In Educational Data Mining 2009.

Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. Decision Support Systems, 49(4), 498-506.

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In Advances in Neural Information Processing Systems (pp. 1019-1027).

Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722.

Goldstein, A., Oestreicher-Singer, G., & Barzilay, O. (2017). Deep into the Funnel? Predicting Online Conversion Using Search Diversity.

Gündüz, Ş., & Özsu, M. T. (2003, August). A web page prediction model based on click-stream tree representation of user behavior. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 535-540). ACM.

Guo, Q., & Agichtein, E. (2010, July). Ready to buy or just browsing?: detecting web searcher goals from interaction data. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 130-137). ACM.

Guo, B., Zhang, R., Xu, G., Shi, C., & Yang, L. (2015, July). Predicting Students Performance in Educational Data Mining. In Educational Technology (ISET), 2015 International Symposium on (pp. 125-128). IEEE.

ICEF. (2015). The state of international student mobility in 2015. Retrieved from http://monitor.icef.com/2015/11/the-state-of-international-student-mobility-in-2015/

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

Kabakchieva, D. (2013). Predicting student performance by using data mining methods for classification. Cybernetics and Information Technologies, 13(1), 61-72.

Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. Knowledge-Based Systems, 23(6), 529-535.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Li, X., Wang, T., & Wang, H. (2017, March). Exploring N-gram Features in Clickstream Data for MOOC Learning Achievement Prediction. In International Conference on Database Systems for Advanced Applications(pp. 328-339). Springer, Cham.

Lin, L., Hu, P. J. H., Sheng, O. R. L., & Lee, J. (2010). Is stickiness profitable for electronic retailers?. Communications of the ACM, 53(3), 132-136.

Lo, C., Frankowski, D., & Leskovec, J. (2016, August). Understanding Behaviors that Lead to Purchasing: A Case Study of Pinterest. In KDD (pp. 531-540).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1.

Papamitsiou, Z. K., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. Educational Technology & Society, 17(4), 49-64.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In Advances in Neural Information Processing Systems (pp. 505-513).

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. Expert systems with applications, 33(1), 135-146.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(6), 601-618.

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., & Ventura, S. (2013). Web usage mining for predicting final marks of students that use Moodle courses. Computer Applications in Engineering Education, 21(1), 135-146.

Salehinejad, H., & Rahnamayan, S. (2016, December). Customer shopping pattern prediction: A recurrent neural network approach. In Computational Intelligence (SSCI), 2016 IEEE Symposium Series on (pp. 1-6). IEEE.

Scheuer, O., & McLaren, B. M. (2012). Educational data mining. In Encyclopedia of the Sciences of Learning (pp. 1075-1079). Springer US.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Tamhane, A., Arora, S., & Warrier, D. (2017, May). Modeling Contextual Changes in User Behaviour in Fashion e-Commerce. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 539-550). Springer, Cham.

Tang, S., Peterson, J. C., & Pardos, Z. A. (2016, April). Deep Neural Networks and How They Apply to Sequential Education Data. In Proceedings of the Third (2016) ACM Conference on Learning@ Scale (pp. 321-324). ACM.

Webb, G. I., Pazzani, M. J., & Billsus, D. (2001). Machine learning for user modeling. User modeling and user-adapted interaction, 11(1), 19-29.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, 26(2), 13-23.

Wolfswinkel, J. F., Furtmueller, E., & Wilderom, C. P. (2013). Using grounded theory as a method for rigorously reviewing literature. European Journal of Information Systems, 22(1), 45-55.

7. Appendices

Appendix A

This appendix includes more detailed information on the inclusion and exclusion criteria for each set of keywords in the literature search. Most keywords provided papers that were a great starting point. Because the research field of deep learning has developed itself only recently there are not that many standardized terms yet. Therefore, many papers used in this research were found through forward and backward citations.

marketing AND (forecast OR predict)
conversion AND (forecast OR predict)
Both these combinations of keywords were used to search for studies that looked at predictive modelling in a marketing context. The second keyword specifically focus on conversion where the first keyword also finds studies that focus on predicting purchases, brand awareness etc.

marketing AND ("machine learning" OR "deep learning")
The goal of this combination of keywords is to find examples of studies in marketing that applied machine learning or deep learning.

"Educational data mining"
The goal of using this keyword was to find literature reviews on the topic of educational data mining. These reviews provide a general understanding of the field and its recent developments. Additionally, the studies discussed in the review are a starting point for finding more relevant papers related to this research. Because the search is rather general it was limited to papers published after 2007 with more than 20 citations.

"educational data mining" AND ("machine learning" OR "deep learning")
This combination was used to find studies that compared machine learning models in the field of educational data mining. These studies provide a starting point for the comparison of machine learning models in this study.

deep learning
A general search into the most quoted literature on deep learning. Only papers with more than 20 citations were included in the search. These papers generally discuss the potential of deep learning along with its best practices.

Appendix B

The concept matrix created during the literature search is included below in figure B-1.

The following abbreviations are used in the concept matrix.
TML = Traditional Machine Learning
ANN = Artificial Neural Network
Tree = Tree-based methods like Decision Trees, Random Forest and Gradient Boosted Trees models
SVM = Support Vector Machine
LR = Logistic Regression
Bayes = Bayesian methods
k-NN = k-Nearest Neighbors
DNN = Deep Neural Network
CNN = Convolutional Neural Network
RNN = Recurrent Neural Network

Articles	Type of Paper =		TML or DL used =		Focus of variable in marketing study =					Focus of EDM study =			Traditional Machine Learning Models used =						Deep Learning Models used =		
	Literature Review	Empirical	TML	Deep Learning	Page views	Time	Sequence	Page type	Scrolling	Student Performance	Student Retention	General	ANN	Tree	SVM	LR	Bayes	k-NN	DNN	CNN	RNN
Predicting Conversion																					
Bellman et al. (1999)		X			X																
Lin et al. (2010)		X				X															
Goldstein et al. (2017)		X	X					X						X							
Gündüz & Özsu (2003)		X				X	X														
Chan et al. (2014)		X	X					X									X				
Lo et al. (2016)		X					X	X													
Guo & Agichtein (2016)		X							X												
EDM																					
Romero & Ventura (2007)	X											X									
Romero & Ventura (2010)	X											X									
Scheuer & McLaren (2012)	X											X									
Papamitsiou & Economides (2014)	X											X									
EDM + Machine Learning																					
Romero et al. (2013)		X	X							X			X	X	X			X			
Dekker et al. (2009)		X	X								X			X		X	X				
Kabakchieva (2013)		X	X							X				X			X	X			
Kotsiantis et al. (2010)		X	X							X				X	X		X	X			
Alper & Çataltepe (2012)		X	X							X					X	X	X				
Delen (2010)		X	X								X		X	X	X	X					
Li et al. (2017)		X	X							X						X					
Deep Learning																					
Guo et al. (2015)		X	X	X						X			X		X		X		X		
Piech et al. (2015)		X	X	X													X				X
Tang et al. (2016)		X		X																	X
Tamhane et al. (2017)				X																	X
Srivastava et al. (2014)	X			X																	
Mikolov et al. (2013)	X	X		X																	X
Goldberg & Levy (2014)	X			X																	X

Figure B-1. Concept matrix.

Appendix C

A Random Forest model built as a test showed that this model was extremely biased by the number of page views. This model used less feature extracted input than the final dataset used in this thesis (see figure C-2). The Random Forest Classifier used a balanced dataset with 10.000 converters and 10.000 non-converters. A train-test split of 70% and 30% was used. The model achieved 94% accuracy. Figure C-1 showed that many of the non-converters had 0 official page views. A user needs to have a page view on a study page before they can have the possibility of clicking on a referral button. Figure C-2 shows how the page view feature dominates as by far the most important feature in predicting conversion in this model. Concluding, it is important to consider what you put into the model. Therefore, only users with at least three page views are included in all further models in this thesis.

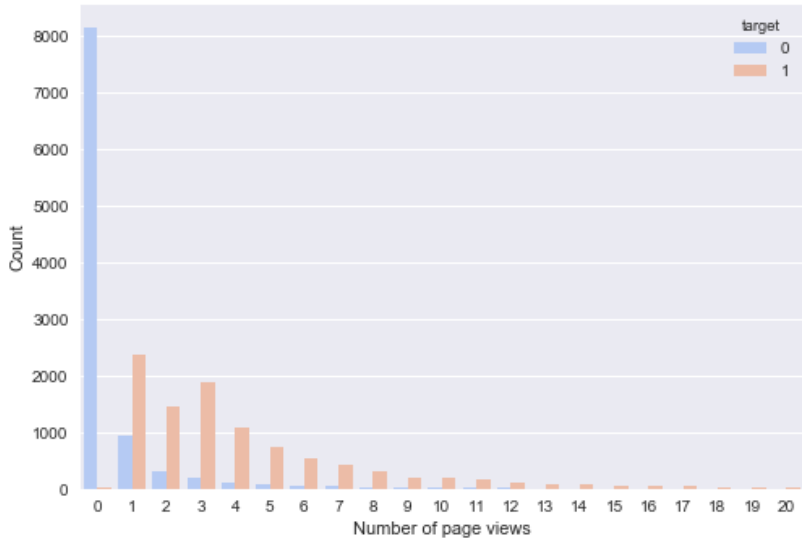

Figure C-1. Count plot of the number of users categorized by their number of page views.

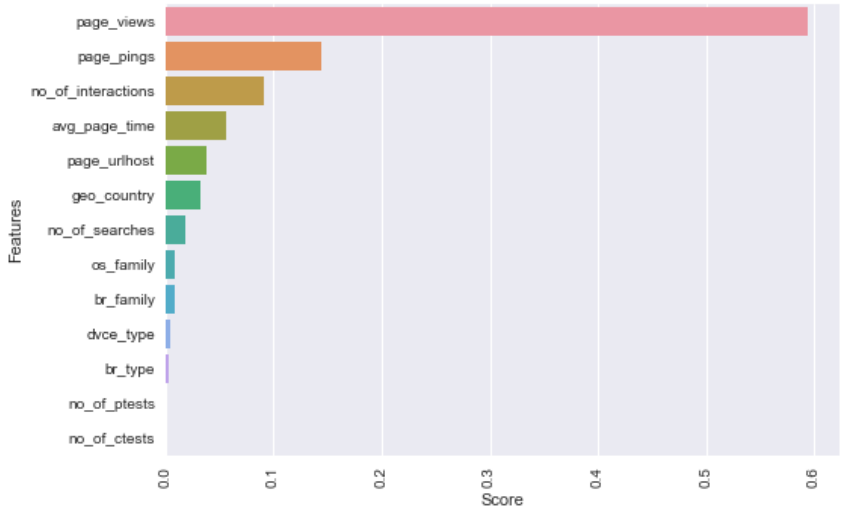

Figure C-2. Feature Importance plot of the Random Forest Model.